



中国研究生创新实践系列大赛
“华为杯”第十八届中国研究生
数学建模竞赛

学 校 东南大学

参赛队号 21102860270

1. 钟丁媛

队员姓名 2. 聂杰文

3. 王惜凡

中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生

数学建模竞赛

题 目 干扰条件下基于多模型协同的 UWB 精确定位

摘 要：

近年来，随着无线设备和传感技术的发展，室内定位技术成为研究热点。超宽带(UWB, Ultra-Wideband) 技术是一种脉冲无线通信技术，具有强穿透力、高传输速率、高时间分辨率等优势，是最具发展前景的室内定位技术之一。尽管如此，UWB 在室内定位的实际环境应用中仍存在诸多问题。环境因素导致的误差和实验软硬件引起的测距误差严重影响着 UWB 滤波的稳定性，降低了定位精度，这造成 UWB 在现实场景中的应用限制，成为该技术发展的重要限制因素。

基于此，本文旨在构建 UWB 精确定位模型，采用高斯牛顿迭代法实现初步定位。考虑到实际测量中的测量噪声和发生遮挡干扰造成的非视距(NLOS) 误差，在预定位的基础上进行校正补偿，由于 NLOS 误差显著高于噪声误差，因此本文决定采用决策树模型首先对数据点进行异常判断分类，同时构建了两个针对有无遮挡干扰的专用 BP 神经网络，用于误差补偿。此外，本文提出了一种归一化方法，有效地提高了模型的跨场景鲁棒性。另一方面，本文将该精确定位模型推广到了靶点运动轨迹的动态跟踪中，在对每个数据点单独完成精确定位的基础上，应用拓展卡尔曼滤波(EKF) 将前后数据点的时间关联性纳入考虑，并利用 Savitzky-Golay (SG) 平滑处理使动态轨迹更加贴合实际。

针对问题一，要求将数据转化为二维表形式后，进行数据清洗。本文提取的数值为各组样本中 4 个锚点的测距值，对题中所提出的潜在无用数据的操作为：剔除异常值和相同值、补全缺失值、保留相似值。经处理，剔除正常数据中 1108 组异常值、11094 组相同值，剔除异常数据中 1459 组异常值、3946 组相同值，最后保留 65008 组“正常数据”样本、73799 组“异常数据”样本。其中，对“正常数据”的第 24 个数据文件保留 134 组样本、第 109 个数据文件保留 220 组样本，对“异常数据”的第 1 个数据文件保留 336 组样本、第 100 个数据文件保留 263 组样本。

针对问题二，要求分别对“正常数据”和“异常数据”构建 UWB 精确定位模型，并计算模型的有效性。基于 BP 神经网络算法，对“异常数据”构建异常融合补偿模型、对“正常数据”构建误差校正模型，分别训练，将“异常”转化为“正常”处理，将“正常”去测量误差处理。针对坐标点的计算误差，利用最小二乘法求解关于距离的非线性方程组的数值解，并利用 Caffery 方法和 Caffery-Talyor 方法进一步优化，考虑到测量误差与实际距离有关，因此引入高斯牛顿梯度下降法进行二次优化，得到靶点坐标的最终估计值。基于以上模型，对附件 2 测量数据进行精确定位，实现对“正常数据”13.51cm 的三维误差、4.58cm 的二维误差，并分别在 XYZ 轴实现 2.90cm、2.90cm 以及 12.03cm 的误差精度；实现对“异常数据”41.40cm 的三维误差、18.74cm 的二维误差，并分别在 XYZ 轴实现 12.73cm、12.32cm 以及 34.91cm 的误差精度。

针对问题三，要求将前一问的模型运用到实验场景 2 中，以此证明该定位模型的应用

广泛性。计算得到附件三 10 组数据的靶点定位，并根据对正常数据的距离回测，实现平均 18.26cm 的误差精度，最小误差达到 0.47cm，仍处于较好的定位水平，验证了模型的可拓展性。

针对问题四，要求建立分类模型，以对数据进行“正常”和“异常”判断。本文采用决策树模型进行信号干扰的检测，通过对附件 1 的特征提取学习，实现对信号存在干扰和不存在干扰的正确检测率分别为 92.1%和 98.2%，平均准确率达到 95.1%。对于附件四 10 组数据的分类为，信号存在干扰：数据 2、数据 3、数据 4、数据 6、数据 8 和数据 9 共六组；信号不存在干扰：数据 1、数据 5、数据 7 和数据 10 共四组。

针对问题五，要求在前文静态精确定位模型的基础上，根据靶点自身运动规律，对靶点的运动轨迹进行跟踪和监测。此问题可视为时间间隔足够小时，静态模型在每个离散时间节点的状态叠加。通过对 549 个数据样本进行预处理，得到 500 个剩余样本，对剩余样本进行决策树信号干扰检测，发现有 219 个“异常数据”。基于此，建立了 EKF 滤波预测模型进行轨迹预测，考虑到信号干扰的随机性和不可避免性，将此模型进一步优化，构建了自适应 EKF 滤波模型。并通过 SG 平滑处理，得到最终完整的运动轨迹图。

本文构建的基于决策树信号干扰分类的 BP-GN 精确定位优化模型能适用于不同规格的室内定位问题，且定位准确率高。建立了基于 SG 平滑优化的自适应 EKF 滤波预测模型，能够实现对定位点运动轨迹的动态跟踪。该模型可广泛推广到实际生活中，在智慧养老、网络故障定位等现实问题解决中均能发挥显著作用。

关键词：异常判断、融合补偿校正、BP 神经网络、高斯牛顿迭代法、决策树分类、自适应 EKF 滤波、跨场景定位

目 录

1 问题重述	4
1.1 问题背景.....	4
1.2 问题提出.....	5
2 模型假设	6
2.1 基本假设.....	6
2.2 符号说明.....	6
3 问题一求解	7
3.1 任务一问题分析.....	7
3.2 任务一求解.....	8
4 问题二建模与求解.....	11
4.1 任务二问题分析.....	11
4.2 基于 BP-GN 的精确定位优化模型.....	12
4.2.1 高斯牛顿迭代法 (GN) 优化定位法.....	12
4.2.2 基于 BP 神经网络的距离补偿法.....	13
4.2.3 基于 BP-GN 的有效性判断.....	15
4.3 模型的求解.....	17
4.3.1 定位模型的有效性评估.....	17
4.3.2 附件二的 Tag 定位.....	18
5 问题三建模与求解.....	19
5.1 任务三问题分析.....	19
5.2 任务三求解.....	20
6 问题四建模与求解.....	21
6.1 任务四问题分析.....	21
6.2 任务四模型建立.....	21
6.2.1 决策树模型介绍.....	21
6.2.2 基于决策树的信号干扰分类模型.....	22
6.3 任务四模型求解.....	24
6.3.1 过拟合检验.....	24
6.3.2 分类模型的准确性.....	24
6.3.3 附件 4 的信号干扰检测.....	25
7 问题五建模与求解.....	26
7.1 任务五问题分析.....	26
7.2 基于自适应 EKF 滤波的定位预测模型	27
7.2.1 具有数据缺失的动态定位问题.....	27
7.2.2 SG 平滑处理	29
7.3 任务五模型求解.....	30
8 模型评价与推广	32
8.1 模型评价.....	32
8.1.1 模型的优点.....	32
8.1.2 模型的缺点.....	32
8.2 模型推广	32
参考文献	33
附录	34

1 问题重述

1.1 问题背景

近年来,随着无线设备和传感技术的发展,室内定位技术成为研究热点,广泛应用于人们的生活场景中。超宽带(UWB, Ultra-Wideband) 技术是一种脉冲无线通信技术,能够通过纳秒甚至微秒级的非正弦波窄脉冲进行数据传输,具有很宽的频谱范围,按照美国联邦通信委员会 FCC 的规定,其频率范围包括从 3.1GHz 到 10.6GHz 之间。正因其技术特征,UWB 技术有着较强的穿透力、快速的传输速率、较高的时间分辨率、良好的抗多径干扰能力等优势,在军事、物联网、交通等各个领域都有着广阔的应用空间,成为最具发展前景的室内定位技术之一^[1]。

UWB 一般通过基于测距的方式达到高精度定位。在采用 UWB 定位时,实验区域内放置多个锚点,锚点与 Tag 通过超宽带信号进行通信,通过超宽带信号的到达情况估计用户与锚点的距离。常见的 UWB 室内定位方法包括:基于飞行时间(TOF)、基于飞行角度(AOA)、基于飞行时间差(TDOA)等。基于飞行时间(TOF)的测距原理是最常用的 UWB 室内定位方法之一,目前 TOF 测量的类型包括:单向测距、双向测距(TWR)和私有。根据 IEEE 802.15.4a 标准^[3],双向测距(TWR)是物联网服务的必选类型。TWR 是指跟踪对象和每个参考点之间进行双向消息交换,在 TWR 机制中,参考点首先向跟踪对象发送信号并记录出发时间,当跟踪对象接收到来自参考点的信号时,跟踪对象将发送回复信号,参考点根据信号出发时间和回复到达时间之间计算信号飞行时间,将其乘以空气中恒定的光速,来估计自己与跟踪目标之间的距离。基于此距离信息,室内位置跟踪系统可以通过定位方法确定跟踪目标的位置。在 TOF 方法中,一维定位问题只需一个基站,二维定位问题需要三个及以上的测距基站,但特殊模式下能够和两个基站测距,三维定位问题则需要四个基站进行测距。

在 2002 年以前,UWB 技术只能在军方使用。2002 年 2 月,美国联邦通信委员会 FCC 正式批准 UWB 民用,UWB 技术才正式走向多种应用场景。例如,“智寻”实时定位系统,提供基于 UWB 技术的定位解决方案,通过 UWB 技术超精密、低功耗、低费用的特点,开发各种服务模式,并拥有实现这些服务模式的各种方案。“智寻”实时定位系统可以在建筑物内部或一定范围内的开放区域,定位运动物体并获得其移动信息。当人员进入定位区域后,定位基站可以在任何时刻任意位置接收信号,并将其传输至监控中心服务器,通过数据处理得到具体信息,例如人员 ID、位置和具体时间,此外能够将其实时动态显示在监控中心的大屏幕或电脑上,以便于管理人员了解员工的工作状态。

尽管如此,UWB 在室内定位的实际环境应用中仍存在诸多问题,例如多路径效应、非视距(non line of sight, NLOS)传播等。一方面,UWB 系统的定位精度取决于实验软硬件等因素,例如超宽带信号的形状、超宽带信号的 TOA 和用户到锚点的距离估计等。另一方面,除 UWB 信号传播特性的影响外,由于实际室内场景的复杂多变,UWB 的信号受到多种因素的干扰,导致其测量精度的误差。NLOS 误差和实验软硬件引起的测距误差在 UWB 滤波解算过程中均表现为量测信息异常,会严重影响滤波的稳定性和定位精度^[2]。尤其是在受到较强干扰时,测量结果产生的异常波动会导致室内定位的无法进行,在特殊应

用场景下甚至导致严重事故的发生。因此，如何对信号干扰下的 UWB 进行高精度的位置估计，是破除 UWB 在室内定位领域发展瓶颈的关键。

1.2 问题提出

基于上述研究背景，如图 1-1 所示，本文考虑有 4 个 UWB 锚点的三维空间定位问题，根据给定的数据集，解决问题如下：

（1）数据清洗

对附件 1 的 648 个数据文件进行数据抓取，采用每行一组数据的二维表形式。将抓取的数据中异常、缺失、相同或相似的数据进行删除，并以矩阵形式展示 24.正常.txt、109.正常.txt、1.异常.txt、100.异常.txt 经清洗后的保留数据。

（2）定位模型

在数据清洗的基础上，分别对“正常数据”和“异常数据”建立合适的定位模型，以估计 Tag 的精确位置，并说明所建模型的有效性，以及在不同维度上的精度。通过所建立的模型，对附件 2 中“正常”和“异常”各 5 组数据进行三维坐标的精确定位。

（3）不同场景应用

验证模型的可推广性。在新的实验场景下，通过上述建立的定位模型，对附件 3 中“正常”和“异常”各 5 组数据进行精确定位。

（4）分类模型

通过先前清洗的数据，建立分类模型，在不清楚信号干扰的情况下，对数据的“正常”和“异常”进行判断。说明分类模型的有效性，并对附件 4 中的 10 组数据进行“正常”和“异常”判断。

（5）运动轨迹定位

动态定位。通过前文构建的静态定位模型，对附件 5 中的动态靶点采集数据，在信号干扰随机发生的情况下，进行靶点运动规律的总结，画出运动轨迹图。

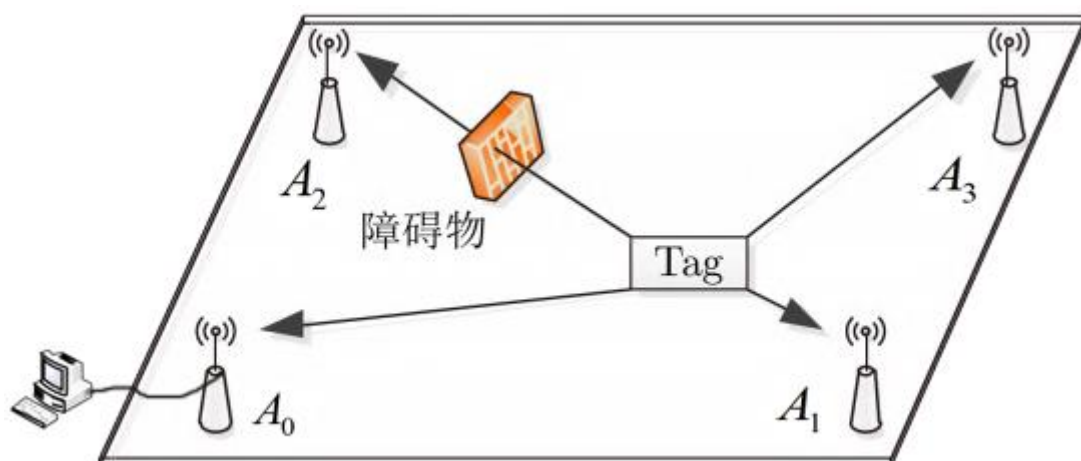


图 1-1 问题实测环境

2 模型假设

2.1 基本假设

在本文的求解过程中，对于问题要求进行细化说明与假设如下：

- (1) 当存在信号干扰时，一组数据中只存在一个异常测距值；
- (2) UWB 的距离测量不存在时间漂移问题；
- (3) 锚点发射信号是以自己为球心，以一定距离为半径球状发射；
- (4) 在信号存在干扰时，仅考虑时间延迟情况下导致的异常波动；
- (5) 锚点和靶点在空间中可视为质点，即忽略运动过程中的姿态变换。

2.2 符号说明

表 2-1 基本符号说明

符号	相关说明
$A_i(x_i, y_i, z_i)$	锚点 i 的坐标($i = 0, 1, 2, 3$)
$P(x_T, y_T, z_T)$	靶点 Tag 的坐标
d_i	靶点到各锚点的距离 $i = 0, 1, 2, 3$
J^T	Jacobi 矩阵的转置
$(\delta_x, \delta_y, \delta_z)^T$	距离在各方向上的测量误差
CD_l	BP 模型中的输入数据, 包含预估靶点坐标及到各锚点的测距值($l = 0, 1, \dots, n$)
\dot{D}_l	BP 模型中的输出数据, 包含 4 个修正的锚点距离值
C	决策树模型中的候选属性集合

3 问题一求解

3.1 任务一问题分析

任务一要求对附件 1 中的 648 个数据文件进行预处理，在转换数据格式为二维表后，对异常、缺失、相同或相似数据进行剔除，并按要求展示部分文件的保留数据。任务一要求的预处理主要包含如图 3-1 所示的两个步骤。

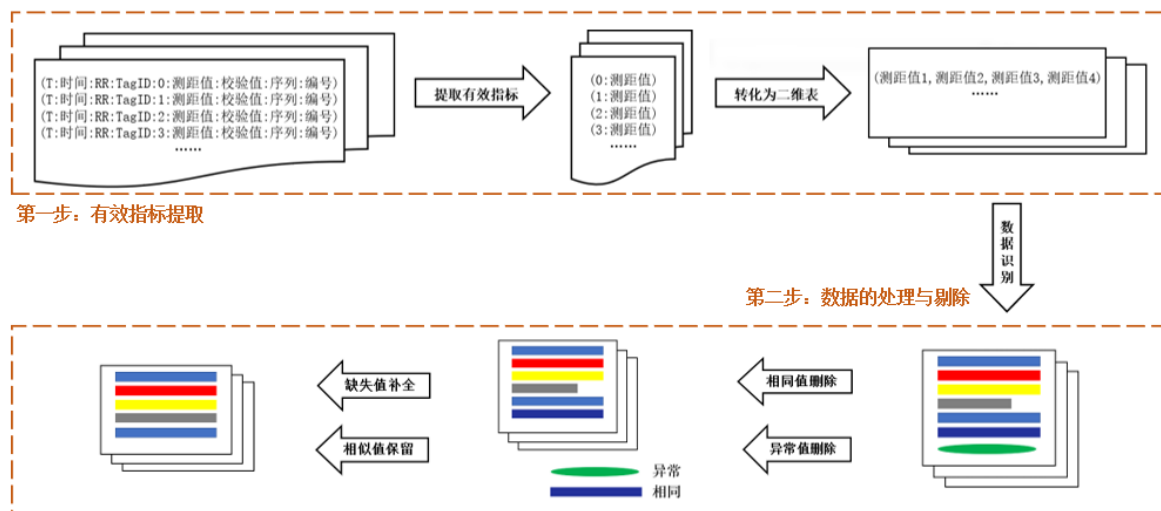


图 3-1 数据预处理步骤

（1）第一步：有效指标提取

在转换二维表格式的过程中，由于题目要求整理为每一行代表一组数据的形式，因此，对于数值的提取应当简明，只提出对后续求解过程有需要的指标。首当其冲，测量数据中最重要的是每个锚点的测距值，是能够定位靶点的主要数据；其次，对于锚点的标号也应当保留，以便于将距离与各锚点坐标对应起来，以进行定位，但在转化成二维表时，可以通过排列进行表示，将每一列对应每个锚点的测距值。因此，我们最后提取的每一组有效数值转化为二维表格式为 (d_0, d_1, d_2, d_3) 。

（2）第二步：数据的处理与剔除

题中对于无用数据的定义为：异常、缺失、相同或相似的数据。根据理解与分析，将其分别处理为：

I.异常值：根据拉依达准则（ 3σ 原则），指一组测定值中与平均值偏差的绝对值大于或等于 3 倍标准偏差的测定值。对于异常值我们进行剔除。

II.缺失值：一组数据中存在测量点数据的缺失。在进行数据的完整性检验后，对存在缺失的数据我们进行插值补全。

III.相同值：对应四个锚点的测距值都相等的样本。对于相同值，我们进行剔除。

IV.相似值：对应四个锚点的测距值相近的样本。对于相似值，考虑到相似的偏差可能由测距误差带来，简单删除可能对于后续的建模造成影响，因此，对于相似值我们进行保留。

3.2 任务一求解

根据上述分析，我们对 4 种潜在无用数据进行的处理数量如表 3-1 所示：

表 3-1 4 种数据的处理数量

处理样本数目	正常数据	异常数据
异常	1108	1459
缺失	0	0
相似	0	0
相同	11094	3946

基于表 3-1，我们并未在附件 1 的样本数据中发现缺失值的存在，因此不做处理；对于相似值我们全数保留，因此不作处理；对于异常值的剔除，“正常数据”和“异常数据”分别剔除了 1108 和 1459 组样本；对于相同值的剔除，“正常数据”和“异常数据”分别剔除了 11094 和 3946 组样本。

经过以上处理后，“正常数据”共计保留了 65008 组样本，“异常数据”共计保留了 73799 组样本。可以知道，“正常数据”和“异常数据”的每个数据文件依然保留了不少的样本，且数量不均。因此，我们通过图 3-2 来展示不存在信号干扰和存在信号干扰下的样本保留数量。

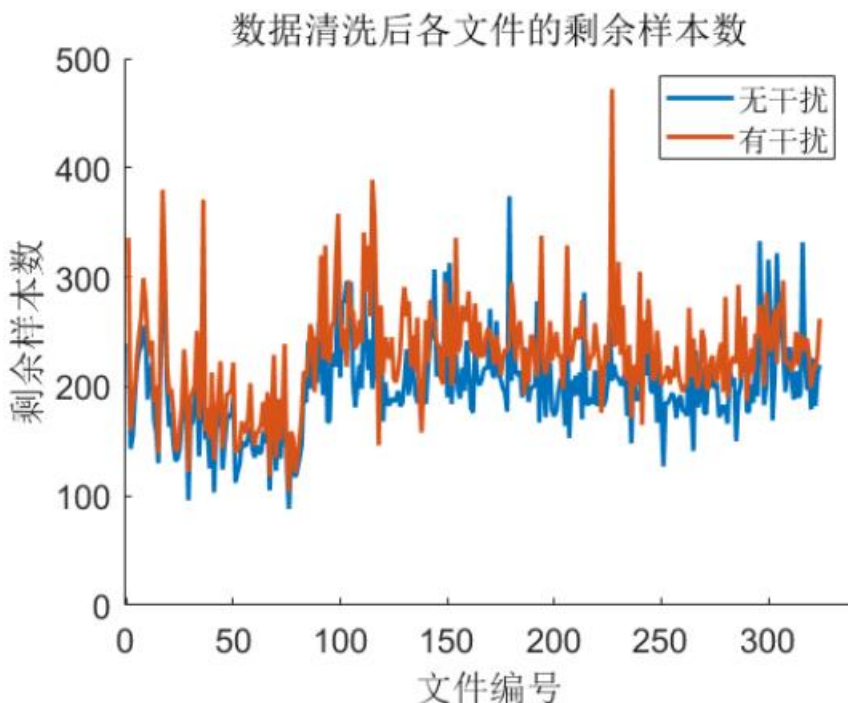


图 3-2 信号存在和不存在干扰时分别保留的样本数量

由图 3-2 可以发现，经过数据的预处理后，数据文件一般的样本保留数量在 100-300 间，信号存在干扰所保留的样本数往往高于信号不存在干扰所保留的样本数。这说明了，

信号存在干扰时，测距值的稳定性将有所降低。

按照任务一要求，需要以 txt 形式展示“正常数据”下的 24 和 109 号数据文件的保留数据，以及“异常数据”下的 1 号和 100 号数据文件的保留数据，这里先给出一个总览，如图 3-3 至图 3-6 所示，具体保留数据将在附件中提交，其中，黄色表示清洗后的保留数据，蓝色表示被清洗的数据。

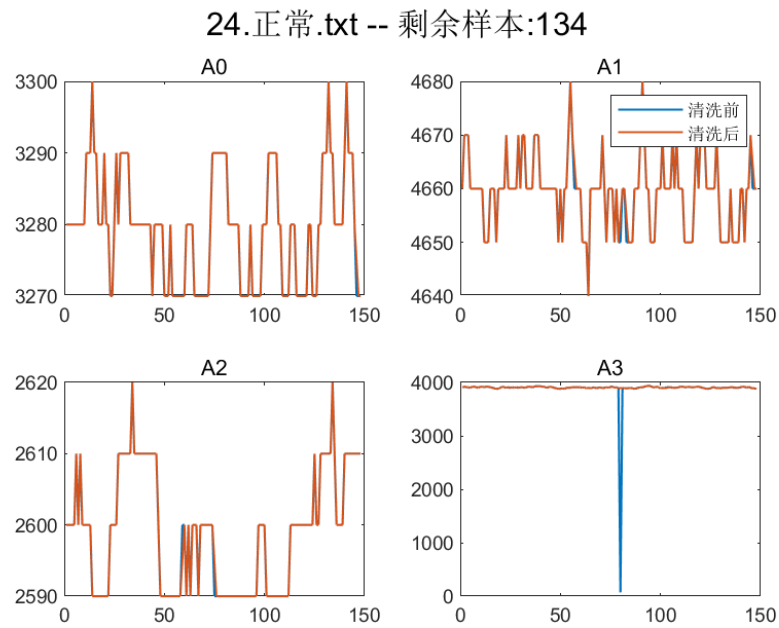


图 3-3 24 号正常文件的保留数据一览

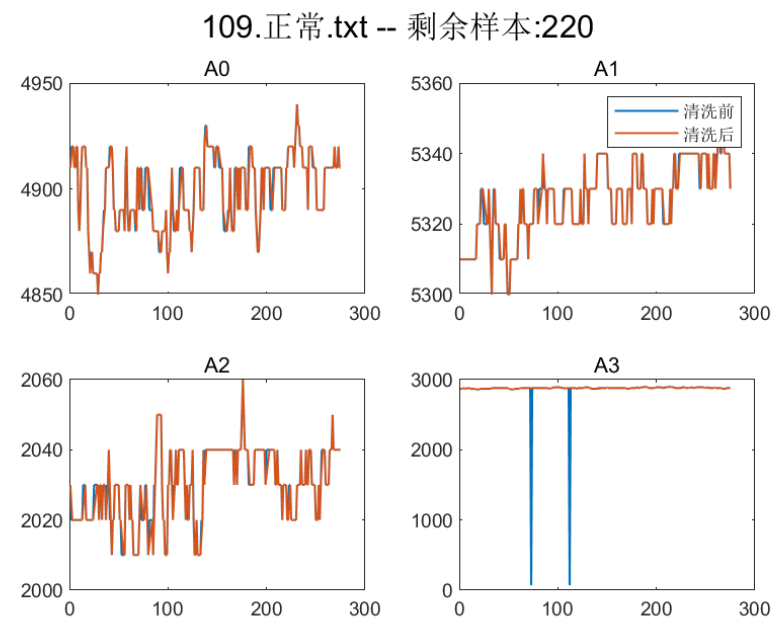


图 3-4 109 号正常文件的保留数据一览

1.异常.txt -- 剩余样本:336

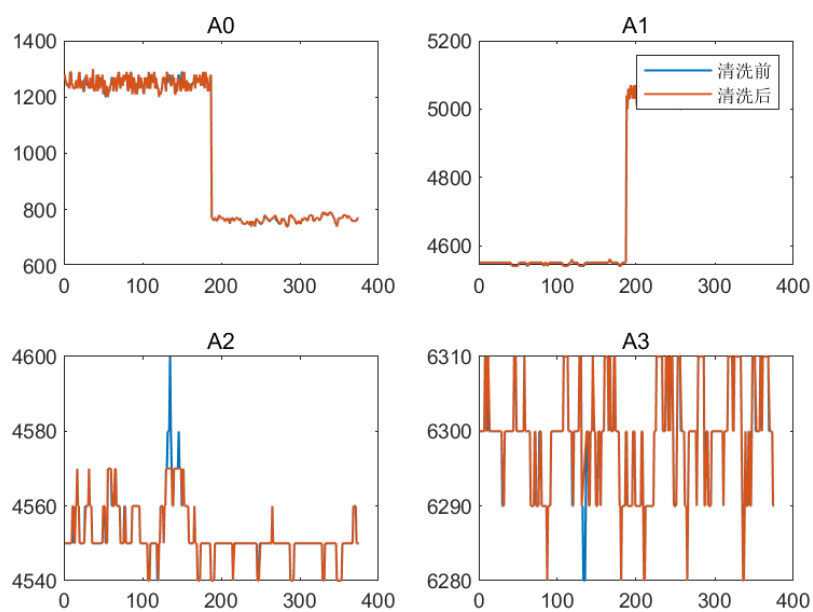


图 3-5 1 号异常文件的保留数据一览

100.异常.txt -- 剩余样本:263

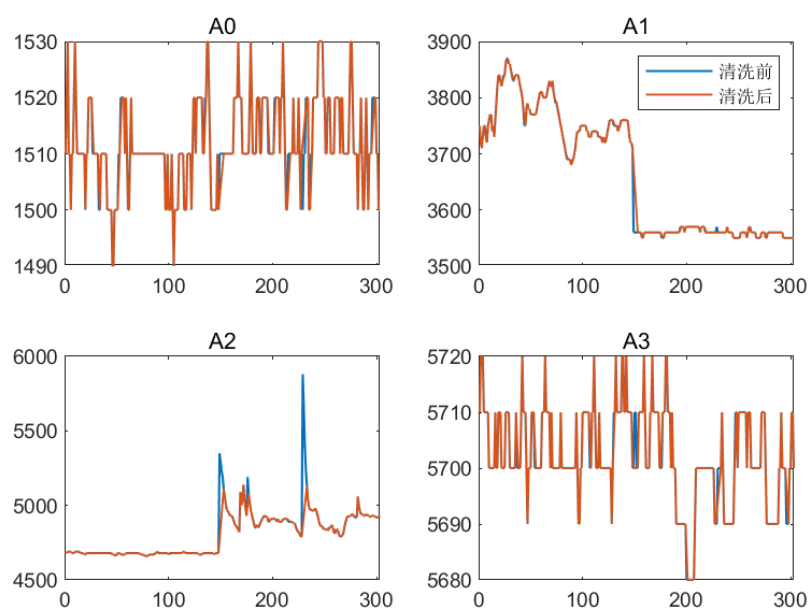


图 3-6 100 号异常文件的保留数据一览

4 问题二建模与求解

4.1 任务二问题分析

任务二要求分别对“正常数据”和“异常数据”构建精确的定位模型，并计算模型的有效性。问题的前提是已知信号是否受到干扰，因此，针对“正常数据”和“异常数据”应当有不同的处理。应当明确的是，“正常数据”是真实距离与测量误差的结果，而“异常数据”则是在“正常数据”基础上存在一个异常测距值。因此，如果能将“异常数据”中的异常测距值进行识别与修正，将其正常化，那么其定位问题则转化为对“正常数据”的测距值。而在“正常数据”的情况下，测量误差仍然是无法避免的问题，如果能够对此进行修正，问题则转化为对准确测距值的精确定位。而即使是已知准确的测距值，虽然在理论上应当得到一个精确的靶点坐标，但在实际计算时仍然因计算方法存在误差。

因此，本问要解决的关键步骤为：

(1) 1组“异常数据”中存在1个异常测距值和3个正常测距值，需要对异常测距值进行识别与修正。

(2) UWB 本身存在的测量误差识别与修正。

(3) 根据4个锚点坐标和对应的距离值，实现1个靶点坐标的精确定位。

根据以上分析，任务二的解题思路如图4-1所示：

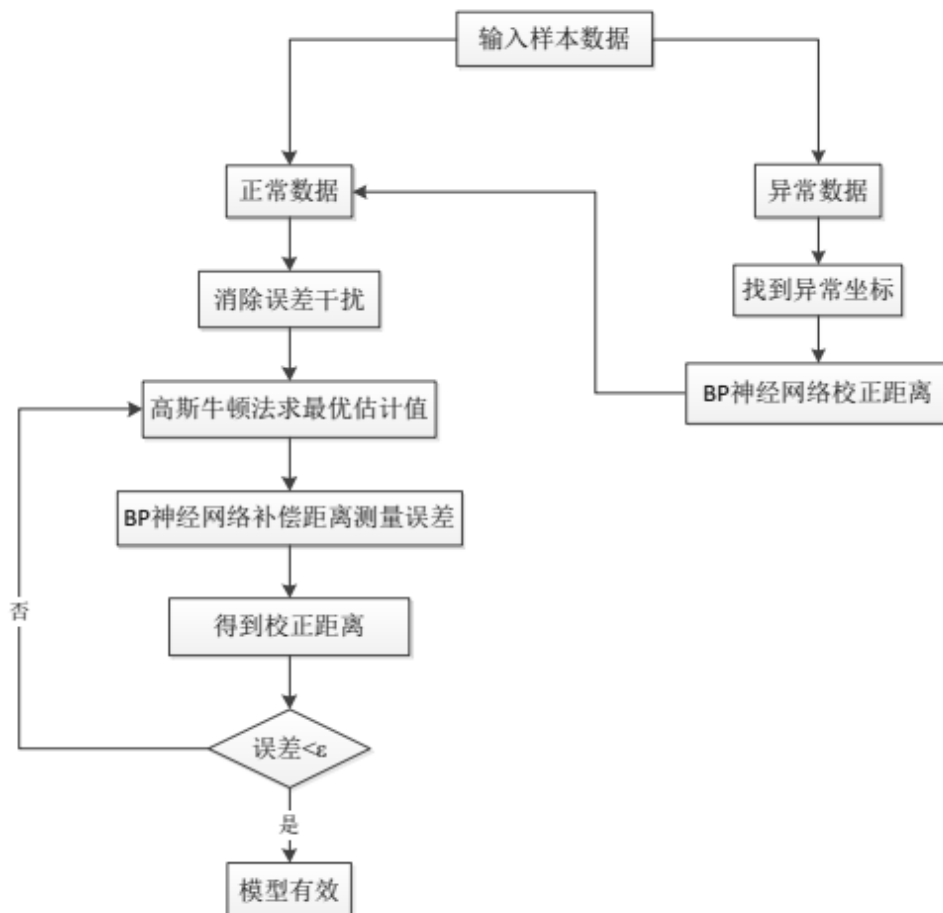


图 4-1 模型定位处理流程

4.2 基于 BP-GN 的精确定位优化模型

4.2.1 高斯牛顿迭代法 (GN) 优化定位法

从几何角度考虑，当不考虑距离的测量误差时，以各个锚点为球心，以距离的测量值为半径的各个球面一定有共同的交点。利用不在同一平面的 4 个锚点对靶点定位，设锚点 i 与靶点 P 间的距离为 d_i ，则 4 个锚点和靶点间的距离应满足：

$$\begin{cases} (x_0 - x_T)^2 + (y_0 - y_T)^2 + (z_0 - z_T)^2 = d_0^2, \\ (x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2 = d_1^2, \\ (x_2 - x_T)^2 + (y_2 - y_T)^2 + (z_2 - z_T)^2 = d_2^2, \\ (x_3 - x_T)^2 + (y_3 - y_T)^2 + (z_3 - z_T)^2 = d_3^2. \end{cases} \quad (4.1)$$

式(4.1)是一个关于 x_T, y_T, z_T 的非线性方程组，很难直接求出此方程组的精确解。同时由于距离存在测量误差，导致各个球面无法精准地相交于同一点，因此靶点定位问题就转换成了求该非线性方程组解的近似解问题，进一步转化为非线性优化最优估计问题。

Caffery 方法

将式(4.1)中的第 i 个等式与第 $i-1$ 个等式相减 ($i=1,2,3$)，得到关于靶点 $P(x_T, y_T, z_T)$ 的一个线性方程组：

$$AX = \mathbf{b} \quad (4.2)$$

其中，

$$A = \begin{bmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_2 & y_3 - y_2 & z_3 - z_2 \end{bmatrix}, X = \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix}, \mathbf{b} = \frac{1}{2} \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 - x_0^2 - y_0^2 - z_0^2 + d_0^2 - d_1^2 \\ x_2^2 + y_2^2 + z_2^2 - x_1^2 - y_1^2 - z_1^2 + d_1^2 - d_2^2 \\ x_3^2 + y_3^2 + z_3^2 - x_2^2 - y_2^2 - z_2^2 + d_2^2 - d_3^2 \end{bmatrix}.$$

利用最小二乘法对式(4.2)求解，可得到该方程的最小二乘解为：

$$X = (A^T A)^{-1} A^T \mathbf{b} \quad (4.3)$$

Caffery-Talyor 方法

利用 Caffery 方法可以计算得出靶点的坐标初值 (x_T, y_T, z_T) ，但由于测量过程中及机器本身存在一定的误差，因此考虑利用循环迭代消除误差的干扰，具体步骤如下模糊距离筛选：

(1) 设 P 点的真值，即明确的靶点位置为 (x_{T0}, y_{T0}, z_{T0}) ，误差为 $\delta = (\delta_x, \delta_y, \delta_z)^T$ ，则有

$$(x_{T0}, y_{T0}, z_{T0})^T - (x_T, y_T, z_T)^T = (\delta_x, \delta_y, \delta_z)^T$$

(2) 式(4.1)中，在 P 点初值处进行一阶 Talyor 展开，忽略二次即以上的项，则

$$M_1 \delta = M_2 \quad (4.4)$$

其中，

$$M_1 = \begin{bmatrix} x_T - x_0 & y_T - y_0 & z_T - z_0 \\ x_T - x_1 & y_T - y_1 & z_T - z_1 \\ x_T - x_2 & y_T - y_2 & z_T - z_2 \\ x_T - x_3 & y_T - y_3 & z_T - z_3 \end{bmatrix}, M_2 = \frac{1}{2} \begin{bmatrix} d_0^2 - (x_T - x_0)^2 - (y_T - y_0)^2 + (z_T - z_0)^2 \\ d_1^2 - (x_T - x_1)^2 - (y_T - y_1)^2 + (z_T - z_1)^2 \\ d_2^2 - (x_T - x_2)^2 - (y_T - y_2)^2 + (z_T - z_2)^2 \\ d_3^2 - (x_T - x_3)^2 - (y_T - y_3)^2 + (z_T - z_3)^2 \end{bmatrix}.$$

利用最小二乘法对式(4.4)求解，可得到该方程的最小二乘解为：

$$\delta = (M_1^T M_1)^{-1} M_1^T M_2 \quad (4.5)$$

(3) 记误差函数 $f(x)$ 为：

$$f(\delta) = \|\delta\|_2^2 = |\delta_x|^2 + |\delta_y|^2 + |\delta_z|^2 \quad (4.6)$$

若 $f(\delta)$ 足够小，则停止迭代；否则，用 $(x_T, y_T, z_T)^T + (\delta_x, \delta_y, \delta_z)^T$ 代替上一次的估计值，转步骤 2 迭代，最后得到最终估计值 $(x_T, y_T, z_T)^T$ 。

GN 迭代法

Caffery-Talyor 方法对所有的测量数据同等看待，无法根据距离不同进行筛选。通过误差分析可知，当测量距离在某些值时误差较大，在这种情况下此方法失效。因此在 CT 方法的基础上，进一步进行高斯牛顿迭代法。利用高斯牛顿法对锚点定位优化，即求误差函数 $f(\delta)$ 的最小值。

对误差函数 $f(\delta)$ ，寻找增量 $\Delta\delta$ ，对其进行一阶 Talyor 展开。要使 $f(\delta + \Delta\delta)$ 达到最小值，则有

$$\Delta\delta^* = \arg \min f(\delta + \Delta\delta) = \arg \min f(\delta + J(\delta)^T \Delta\delta)$$

对上述表达式求导并转化为线性问题求解，得到：

$$J(\delta)J(\delta)^T \Delta\delta = -J(\delta)f(\delta)$$

取 $H(\delta) = J(\delta)J(\delta)^T$ ， $g(\delta) = -J(\delta)f(\delta)$ ，则转化为求解：

$$H\Delta\delta = g \quad (4.7)$$

其具体步骤如下：

- (1) 输入靶点的坐标初始 (x_T, y_T, z_T)
- (2) 对于第 k 次迭代，求出当前的 Jacobi 矩阵 $J(\delta_k)$ 和误差 $f(\delta_k)$ ；
- (3) 求解增量方程 $H\Delta\delta = g$ ；
- (4) 若 $\Delta\delta$ 足够小，则停止迭代；否则，令 $\delta_{k+1} = \delta_k + \Delta\delta_k$ ，返回第 2 步。

经过以上四个步骤，得到最终的估计值 $(x_T, y_T, z_T)^T$ 。

4.2.2 基于 BP 神经网络的距离补偿法

基于 GN 定位法，能够计算得到基于“正常数据”和“异常数据”的初步靶点坐标。根据距离补偿的需要，构建两个补偿网络：

- (1) 异常测距值的融合补偿网络 BP1

BP1 模型的输入节点为 7，包括：初步预估靶点的 3 个坐标值 x_T^0 、 y_T^0 、 z_T^0 ，以及 4

个测距值 d'_i ， $(i=0,1,2,3)$ 。输出节点为 4，即识别异常点后修正的 4 个校正测距值 \dot{d}_i 。该补偿网络由三层神经网络构成，神经元设置为 20 个，输出传递函数为 Log-Sigmoid 函数，中间层传递函数为 Tan-Sigmoid 函数。

(2) 正常测距值的测距误差补偿网络 BP2

BP2 模型的输入节点为 7，包括：初步预估靶点的 3 个坐标值 x_T^0 、 y_T^0 、 z_T^0 ，以及 4 个测距值 d_i ， $(i=0,1,2,3)$ 。输出节点为 4，即测量误差修正的 4 个测距值 \dot{d}_i 。该补偿网络由三层神经网络构成，神经元设置为 20 个，输出传递函数为 Log-Sigmoid 函数，中间层传递函数为 Tan-Sigmoid 函数。

BP 神经网络能够在输入端和输出端构建非线性的映射关系[4]，算法实现流程如图 4-2 所示：

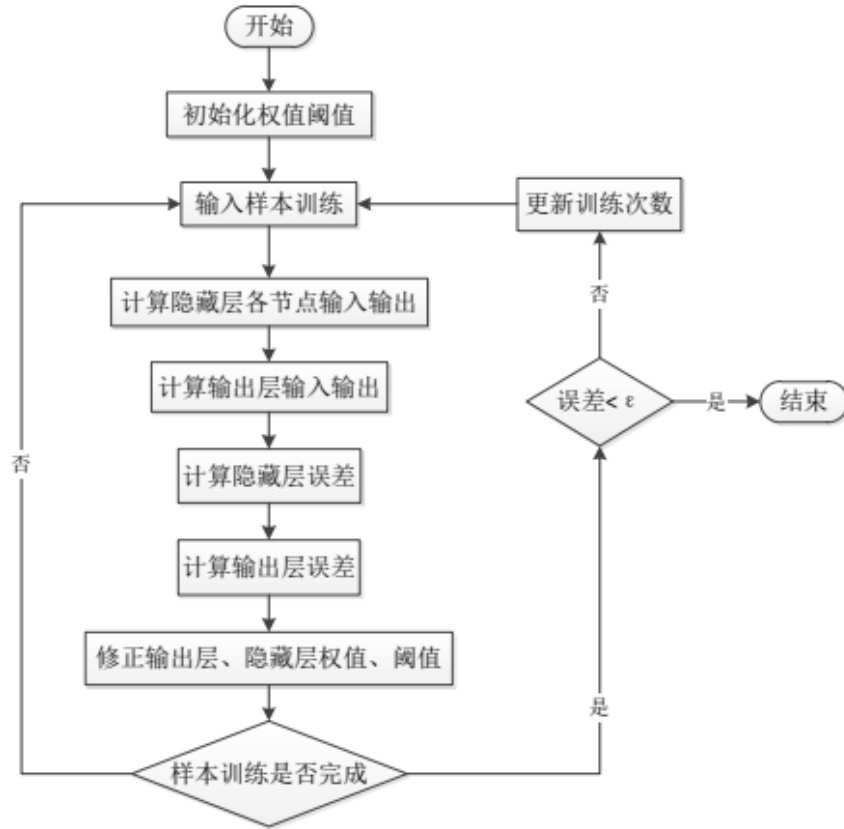


图 4-2 BP 神经网络算法流程

隐含层的神经元个数为：

$$m = \sqrt{n + l} + \alpha \quad (4.8)$$

其中， α 为 1 到 10 之间的常数。

隐含层中第 i 个神经元的输出为：

$$a_{li} = f_1\left(\sum_{j=1}^n w_{ij} p_j + b_{li}\right), i = 1, 2, \dots, m \quad (4.9)$$

输出层第 k 个神经元的输出为：

$$a_{2k} = f_2(\sum_{i=1}^m w_{ki}a_{1i} + b_{2k}), k = 1, 2, \dots, l \quad (4.10)$$

隐含层单元对输入值进行加权求和，从而形成标量的激活，如式(4.11)所示：

$$net_i = \sum_{j=1}^k b_j w_{ij} + w_{i0} = \sum_{j=0}^k b_j w_{ij} \equiv w'_i b \quad (4.11)$$

其中， b 表示增广输入的特征向量， w 表示权向量， i 表示隐含单元的索引。 w_{ij} 表示输入层单元 j 到隐含层单元 i 的权值，也被成为“突触权”。输出距离分量由每个隐含层单元激发，是净激活 net 的非线性函数 $f(net)$ ，即

$$\dot{D}_i = f(net_i) \quad (4.12)$$

神经网络模型图如图 4-3 所示：

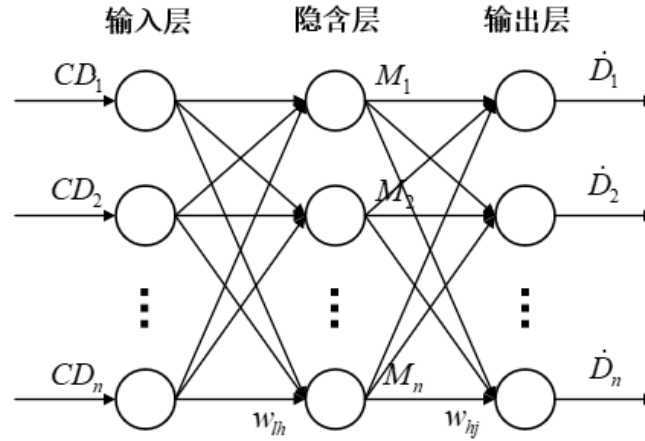


图 4-3 神经网络模型图

4.2.3 基于 BP-GN 的有效性判断

即使经过 GN 的迭代寻优和 BP 的距离补偿修正，误差仍然无法完全避免，其产生的原因来自于：

(1) 测距误差

在 UWB 测距中，基于飞行时间方法的误差主要来自于内部，包括各种典型的情况，例如，即有限带宽、印刷电路板损耗、时钟漂移和热噪声等。多种原因导致晶体振荡器有时并不能准确工作，导致其在跟踪目标和参考点的测量时可能会出现小的正负偏移。由于 TOF 的工作原理，1 纳秒的误差可能导致大约 30 厘米的距离估计误差，因此这个小偏移可能会造成重大影响，这些也是在 UWB 测量时是无法避免的^[5]。

(2) 坐标计算方法误差

由上述的模型构建可知，高斯牛顿的定位优化方法可能存在算法无法收敛的问题，导致局部定位的不准确；基于 BP 神经网络的定位方法则可能存在学习率不稳定、网络性能差等问题，发生定位结果的偏离。以上方法都无法避免地出现计算过程的误差，导致计算得到的靶点坐标始终与真实值存在不同程度的偏差。

对于测量误差，BP 算法可以进行修正；对于计算方法的误差，GN 方法的循环迭代思想可以借鉴。因此，如果能够通过算法流程设计，让能够将 GN 和 BP 进行组合迭代，将能够大大提高定位的估计精度，减小误差对靶点确定的影响。

根据 GN 算法得到第 j 个数据样本的参考 Tag 坐标 $\hat{P}_j(x_{T,j}, y_{T,j}, z_{T,j})$ ，根据参考坐标分别计算参考 Tag 坐标 \hat{P}_j 与锚点 $A_i (i=0,1,2,3)$ 的参考距离值，即 $\hat{d}_{i,j}$ 。比较参考距离值 $\hat{d}_{i,j}$ 与测量值 $d_{i,j}$ 的误差，通过均方根误差(RMSE)来判断数据的异常偏离程度：

$$RMSE = \sqrt{\frac{(\hat{d}_{0,j} - d_{0,j})^2 + (\hat{d}_{1,j} - d_{1,j})^2 + (\hat{d}_{2,j} - d_{2,j})^2 + (\hat{d}_{3,j} - d_{3,j})^2}{4}} \quad (4.13)$$

设定一个误差值 ε ，当 $RMSE < \varepsilon$ 时，认为模型有效，输出定位的 Tag 坐标；否则，当 $RMSE > \varepsilon$ 时，认为模型无效，返回到 GN 的坐标优化计算，进行迭代寻优，直至满足阈值条件为止。

类似地，对于定位模型的有效性判断，需要根据附件 1 的历史数据进行评估，分为以下三个维度：

① 3 维精度

$$XYZMSE = \sqrt{\frac{\sum_{j=1}^n ((X_j - \hat{X})^2 + (Y_j - \hat{Y})^2 + (Z_j - \hat{Z})^2)}{n}} \quad (4.14)$$

② 2 维精度

$$XYMSE = \sqrt{\frac{\sum_{j=1}^n ((X_j - \hat{X})^2 + (Y_j - \hat{Y})^2)}{n}} \quad (4.15)$$

③ 1 维各自精度

$$\begin{cases} XMSE = \sqrt{\frac{\sum_{j=1}^n (X_j - \hat{X})^2}{n}}, \\ Y MSE = \sqrt{\frac{\sum_{j=1}^n (Y_j - \hat{Y})^2}{n}}, \\ Z MSE = \sqrt{\frac{\sum_{j=1}^n (Z_j - \hat{Z})^2}{n}}. \end{cases} \quad (4.16)$$

此外，定义和原始数据相比，本模型的优化效率 η 为：

$$\eta = \frac{|\varepsilon_1 - \varepsilon_2|}{\varepsilon_1} \quad (4.17)$$

其中， ε_1 表示原始数据误差， ε_2 表示本模型误差。

4.3 模型的求解

4.3.1 定位模型的有效性评估

(1) 正常数据

通过构建的定位模型对附件 1 共 65008 组正常样品数据进行 Tag 定位，得到如表 4-1 所示的精度估计（单位：cm，精确到两位小数）。

表 4-1 模型定位正常数据的精度估计

	一维误差			二维误差	三维误差
	X	Y	Z	XY	XYZ
GN	4.10	3.22	31.90	5.77	33.42
BP-GN	2.90	2.90	12.03	4.58	13.51
优化效率	29.27%	9.94%	62.29%	20.62%	59.58%

如表所示，GN 算法得到的误差表示即视为附件 1 原数据中求得的误差值，BP-GN 算法得到的误差表示利用 BP-GN 算法，经过高斯牛顿迭代法和 BP 神经网络的距离补偿法得到的误差值。通过 BP-GN 算法，靶点的定位精度均在一定程度上提高了。

从一维误差来看，由于在原始数据中 x, y 坐标的精度已经很高了，所以优化效率并不高，但基于 BP-GN 模型成功地将精确定位误差范围控制在了 2.9cm 以内。而由于在场景 1 中，4 个锚点的高度差仅为 40cm，而靶点的活动范围高度区间长度为 300cm，因此在靶点活动范围内的绝大部分区间，锚点并不能从 z 坐标上精确定位靶点位置，这便是造成 z 的精度远远不如其他两个的原因，但相比于原数据，本模型将精度提高了 62.29%。

从二维误差来看，在 xy 平面上，能成功将误差范围控制在 4.58cm 以内。从三维误差来看，由于不能从 z 坐标上精确定位靶点位置，因此该模型的误差范围为 13.51cm，但相比于原始数据，精度提高了 59.59%。

因此相比于原始数据，基于 BP-GN 的精确定位模型有效性很强，能在 4 个测量距离值准确的情况下实行精准定位。

(2) 异常数据

通过构建的定位模型对附件 1 共 73799 组异常样品数据进行 Tag 定位，得到如表 4-2 所示的精度估计（单位：cm，精确到两位小数）。

表 4-2 模型定位异常数据的精度估计

	一维误差			二维误差	三维误差
	X	Y	Z	XY	XYZ
GN	16.36	15.47	87.36	23.48	92.16
BP-GN	12.73	12.32	34.91	18.74	41.40
优化效率	22.19%	20.36%	60.04%	20.19%	55.08%

如表所示，由于数据值本身存在异常情况，因此相比于正常值而言，异常值的误差较高，但总的来说，相比于原始情况，该模型在很大程度上提高了定位精度。从一维误差来

看，BP-GN 模型成功地将 x, y 坐标上的精确定位误差范围控制在了 12.73cm 和 12.32cm 以内，将 z 的精度提高了 60.04%。

从二维误差来看，在 xy 平面上，能成功将误差范围控制在 18.74cm 以内。从三维误差来看，将模型的精度提高了 55.08%。

基于上述数据分析，可以得出结论：本部分建立的基于 BP-GN 的精确定位模型有效性很强，无论距离测量值是否存在异常，都能实现精准定位。

4.3.2 附件二的 Tag 定位

根据所建立的 BP-GN 定位模型，对附件二中 10 组数据的定位结果如表 4-3 所示（单位：mm，精确到个位数）。

表 4-3 附件二 10 组数据的靶点定位

	信号无干扰		
	坐标 X	坐标 Y	坐标 Z
数据 1	1158	639	1087
数据 2	3174	1736	847
数据 3	2709	1209	1056
数据 4	2478	1007	1745
数据 5	1503	2508	1881
	信号有干扰		
	坐标 X	坐标 Y	坐标 Z
数据 6	2004	795	1428
数据 7	4175	1699	1511
数据 8	1810	1307	1583
数据 9	3537	1973	1520
数据 10	4668	2104	1604

将附件二中 5 个无干扰样本的观测距离视为准确测距值，通过对比定位结果与 4 个锚点的距离与测距值的差，来判断所构建定位模型对这 5 组正常数据的定位精度，进行模型回测，如表 4-4 所示（单位：cm，精确到两位小数）：

表 4-4 模型对附件二 5 组正常数据的定位精度

	平均误差	最大误差	最小误差
误差值	3.46	9.62	0.08

根据表 4-4 所示，基于 GN-BP 的定位模型对于附件二的 5 组正常数据已经达到平均 3.46cm 的定位精度，最小误差达到 0.08cm，已经达到非常精确的定位效果。

而由于干扰样本并不能够确定干扰锚点及干扰程度，我们的定位模型也只能做到估计，因此没有办法作为回测依据。

5 问题三建模与求解

5.1 任务三问题分析

任务三需要将前文中建立的定位模型应用到不同的场景中，以此体现模型的适用广泛性。在任务二中基于 BP-GN 定位优化模型，分别对信号有无干扰的情况进行分析，对靶点位置进行了精确定位。如图 5-1 所示，任务二中的实验场景 1 靶点范围在一个底面边长为 $5000\text{mm} \times 5000\text{mm}$ 的正方形，高度为 3000mm 的长方体中，而锚点的高度仅在长度为 400mm 的区间中；而本任务中的实验场景 2，如图 5-2 所示，靶点范围在一个侧面边长为 $3000\text{mm} \times 3000\text{mm}$ 的正方形，长度为 5000mm 的长方体中。通过观测可以注意到，在这两个实验场景中，锚点位置均在长方体的 4 条侧棱上，且均未到达顶点。

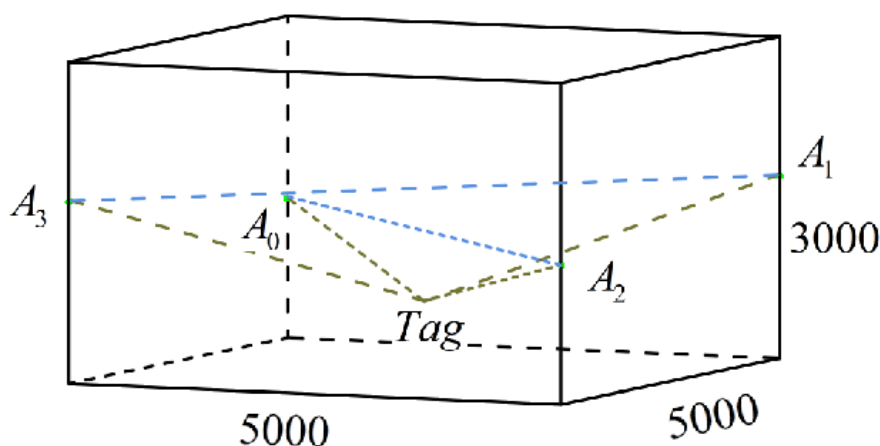


图 5-1 实验场景 1

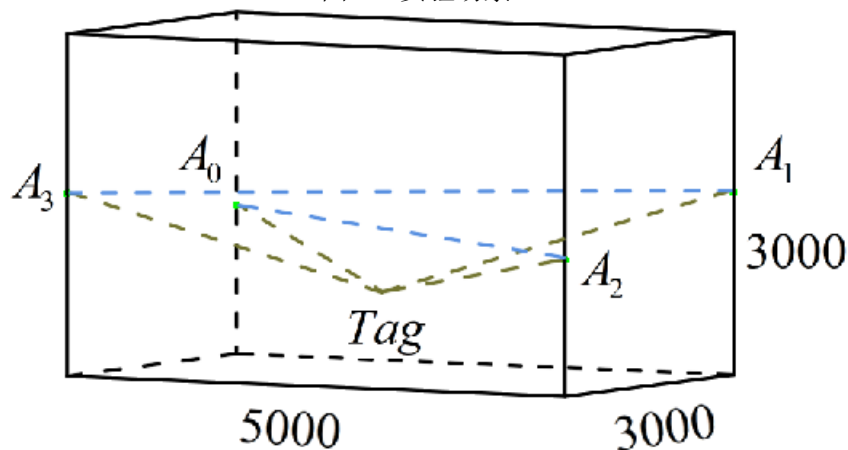


图 5-2 实验场景 2

由题可知，在此问题中各给出了 5 组信号有干扰和无干扰的距离信息，当距离信息受到干扰时，测量距离错误。可以将这 10 组距离信息分别代入任务二中不同的情况进行求解。但需要注意的是，任务二中，通过对已知数据的对比和分析，可以很容易找出测量距离的错误值，即可以明确在哪个锚点产生测量干扰。而在此问题中，无法直接指出错误距离出现在哪个锚点中，因此在有信号干扰的精确定位中，应首先明确错误距离值出现位置。而后利用任务二给出的模型，对错误距离进行校正，转入正常的问题求解中。

5.2 任务三求解

靶点坐标求解

根据任务二所构建的 GN-BP 模型，将附件 3 中的 10 组数据带入，得到的靶点坐标定位如表 5-1 所示（单位：mm，精确到个位数）：

表 5-1 附件三 10 组数据的靶点定位

	信号无干扰		
	坐标 X	坐标 Y	坐标 Z
数据 1	3858	1876	1437
数据 2	4428	1632	1542
数据 3	3315	1696	1448
数据 4	2606	1689	1475
数据 5	209	667	1516
	信号有干扰		
	坐标 X	坐标 Y	坐标 Z
数据 6	4871	1805	1517
数据 7	2458	1550	1358
数据 8	1539	1477	1563
数据 9	1853	1179	1358
数据 10	925	1162	1501

距离回测

将附件三中 5 个无干扰样本的观测距离视为准确测距值，通过对比定位结果与 4 个锚点的距离与测距值的差，来判断所构建定位模型对这 5 组正常数据的定位精度，进行模型回测，如表 5-2 所示（单位：cm，精确到两位小数）

表 5-2 模型对附件二 5 组正常数据的定位精度

	平均误差	最大误差	最小误差
误差值	18.26	70.61	0.47

根据表 5-2 所示，基于 GN-BP 的定位模型对于附件三的 5 组正常数据已经达到平均 18.26cm 的定位精度，最小误差达到 0.47cm。

对比任务二中的表 4-4 可以发现，我们所构建的定位模型在新场景中的定位准确度明显下降，但仍然在合理的范围。这是由于，GN-BP 定位模型的学习特征提取是基于场景 1 的数据，而在场景 2 中，靶点范围及锚点坐标的改变，均与历史的学习特征产生差异。但我们所构建定位模型的效果降低始终在合理的范围，说明 GN-BP 定位模型的稳定性和实用性较好，即使在不同的环境中，仍能够得到较为合理的定位结果。

6 问题四建模与求解

6.1 任务四问题分析

任务四要求建立分类模型，以对数据进行“正常”和“异常”判断。任务二和任务四的定位模型建立在已知数据是否异常的条件上，但实际情形中，信号是否受到干扰是未知的。因此，任务四是前问所构建定位模型的应用前提，即，面对实测数据，首先应当进行信号干扰判断，再根据不同情况代入模型，进行精确定位。基于此目的，本问需要对已知锚点坐标、给定测量数据的情况下，进行是否发生信号干扰的判断。根据前文的求解，对于测量数据仅提取对应锚点的距离值，因此，本问的已知条件为：4 个锚点坐标、4 个对应测距值，需要得到的结果为：是否存在信号干扰（是、否）。

由题意可知，发生信号干扰时，数据发生的异常波动通常表现为时间延时，也即，飞行时间延长，此时会导致测距值大于真实值。根据基本假设（1），当存在信号干扰时，4 个测距值中仅存在 1 个异常值，其余 3 个均为正常测距值。但根据 3 个锚点坐标及对应测距值，得到的是相交于关于锚点平面对称的两点^[6]，这两点到异常测距值对应锚点的距离应当无法与异常测距值匹配。也即，根据异常数据定位得到的异常 Tag 参考坐标，其与锚点距离的计算值应当与测距值存在差异，当不存在误差时，这可以作为异常数据的判断依据。但根据锚点坐标及测距值的 Tag 定位过程本身存在误差，加之实际 UWB 测距本身存在的误差，仅仅将测距值与参考计算距离是否匹配作为判断依据，可能会导致正常数据被误判为异常数据，分类结果的准确性大大降低。因此，本问的难题在于，当存在测量误差和定位模型误差的情况下，如何进行准确的信号干扰判断。

仅仅进行数理分析判断，很容易陷入主观思维的局限性，导致分类判断的不准确。因此，我们考虑采用机器学习算法，通过对附件 1 数据集的干扰特征学习，来进行信号干扰的检测判断。综合比较下，我们选择采用决策树学习，通过特征提取来进行“正常数据”和“异常数据”的分类。

6.2 任务四模型建立

6.2.1 决策树模型介绍

决策树学习是最常用的机器学习方法之一，既适用于数据科学中的分类问题，也适用于回归问题。决策树学习的整体思想是，对整个训练集建立一棵树，并在每片叶子上处理一个单一的结果或使每片叶子上的误差最小化。决策树把最好的特征放在树根上，然后将数据集分成子集，这样的每个子集应该包含具有相同特征值的数据。本问采用决策树作为分类模型的主要原因如下：

（1）决策树能够模仿人类理解和解释数据的思维水平，是一种智能的算法。对于信号干扰边界的判定，决策树能通过测试和学习，提取判断特征。

（2）决策树能够展示数据解释的逻辑，而不是像其他机器学习算法那样采用黑盒算法，例如，支持向量机和神经网络。通过决策树得到信号干扰的判断逻辑，能够更好地总结信号干扰的发生和作用规律，为提高检测技术的精度、现实环境的布置选择提供理论依据。

决策树学习遵循乘积和(SOP)表达式,也即析取范式(DNF)。DNF是对逻辑公式的标准化或规范化,即连接子句的分离。当且仅当一个逻辑公式是字面值的连词析取时,该逻辑公式被认为属于DNF。如果一个DNF公式中的变量在每次连接中恰好出现一次,那么该公式就是完整的DNF。在决策树学习中,对于一个类,从树的根到具有相同类的叶节点的每一个分支都是值的合取(乘积),在该类中结束的不同分支构成析取(和)。但决定将哪个特性放置在树的根或不同层次是一个复杂的步骤。仅仅随机选择任何一个特征作为根并不能解决这个问题,可能会导致不好的结果和低准确度。目前,信息增益和基尼系数是常用的特征选择标准。在使用信息增益时,假设特征是分类的,而在使用基尼指数时,需要假设特征是连续的。

6.2.2 基于决策树的信号干扰分类模型

(1) 信号干扰特征判断过程

我们通过递归决策树方法建立了信号干扰检测的分类模型,如图6-1所示:

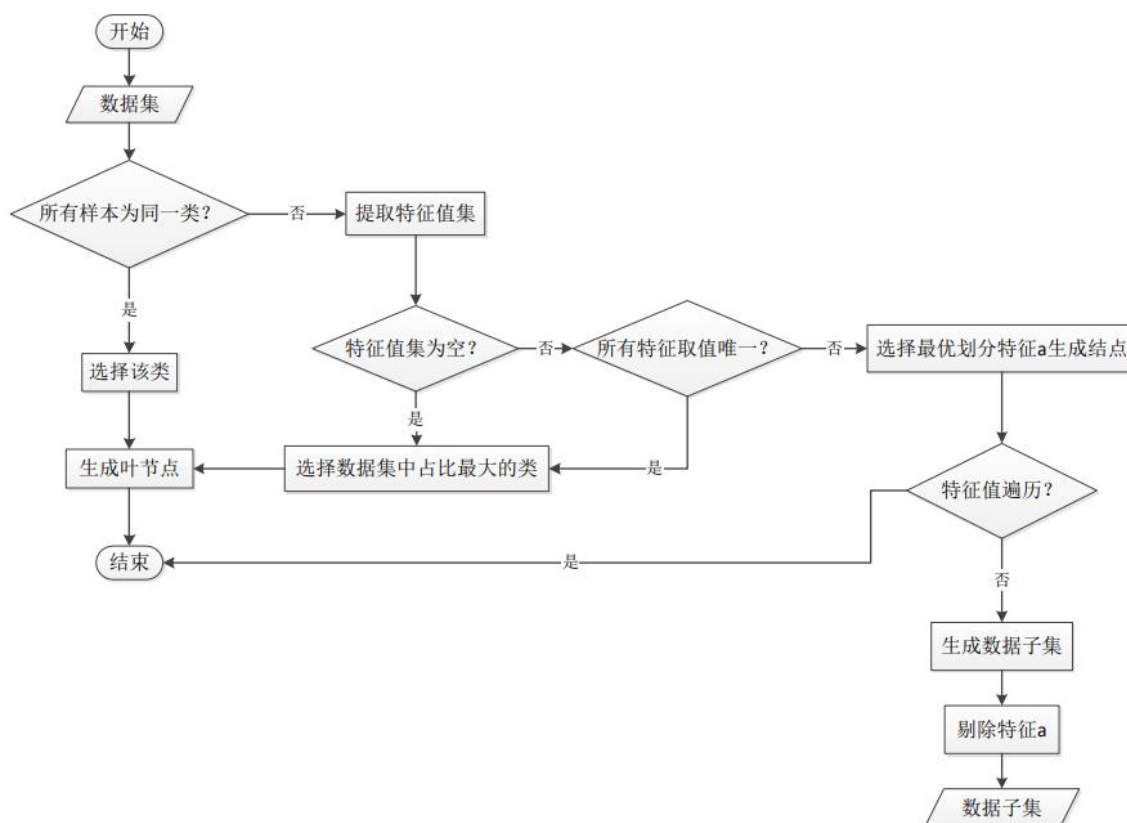


图 6-1 基于决策树的信号干扰建议模型流程图

通过递归分区(RPART)例程包,我们将检测过程分为两个阶段:

第一阶段,将训练数据集分成两组,寻找单一特。将训练数据集递归地分成每个子组,直到这些子组达到精确度水平或无法再改进。

第二阶段,通过交叉验证的方法来检查和估计生成模型的风险。

生成的决策树模型将UWB测量样本分成 k 个终端组,我们为每一个终端组分配一个预测类。式(4.1)说明了分类中,树的某节点 M 对未来观测的概率 P 为:

$$P(M) = \sum_{k=1}^X \pi_k P\{x \in M | \tau(x) = k\} \approx \sum_{k=1}^X \pi_k n_M / n_k \quad (6.1)$$

其中, π_k ($k=1,2,\dots,X$) 是每个类的初始概率, $\tau(x)$ 是 x 的真实分类, x 是预测特征的向量。 n_k 是第 k 个终端组的样本数量, 而 n_M 是节点 M 测距值的数量。

节点 M 以及整个构造树的估计风险由式(6.2)所示:

$$\begin{cases} R(M) = \sum_{k=1}^X p(k|M) L(k, \tau(M)) \\ R(T) = \sum_{l=1}^m P(M_l) R(M_l) \end{cases} \quad (6.2)$$

其中, M_l 是树的终节点, $\tau(M)$ 是 M 最小风险的归类, $L(k, \tau(M))$ 是 M 被错误分类的损失矩阵。此外, 为了计算分裂适应度的杂质, 我们对节点 M 构建了如式(6.3)所示的杂质函数 f :

$$I(M) = \sum_{k=1}^X f(p_{kM}) \quad (6.3)$$

其中, p_{kM} 是节点 M 中属于 k 类的预期样本比例。以精确度为目标, 当 M 纯粹时, $I(M)=0$, 因此 f 应该是 $f(0)=f(1)=0$ 的凹函数。

(2) 属性划分标准——基尼系数

我们采用基尼系数标准 $f(p)=p(1-p)$, 通过减少最大杂质的原则来进行树的拆分。

例如, 节点 M 的纯度可用基尼值来度量:

$$Gini(M) = \sum_{k=1}^X \sum_{k' \neq k} p_{kM} p_{k'M} = 1 - \sum_{k=1}^X p_{kM}^2 \quad (6.4)$$

$Gini(M)$ 反应了从节点 M 中随机抽取两个样本时, 类别标记不一致的概率, $Gini(M)$ 越小, 则节点 M 的纯度越高。

属性 c 的基尼系数定义为:

$$Gini_index(M, c) = \sum_{v=1}^V \left| \frac{M^v}{M} \right| Gini(M^v) \quad (6.5)$$

其中, 假设离散属性 c 存在 V 个可能的取值 $\{c^1, c^2, \dots, c^V\}$, 如果用 c 对节点 M 进行划分, 则会在此基础上产生 V 个分支节点, 其中第 v 个分支节点包含了节点 M 中所有在属性 c 上取值为 c^v 的样本, 记为 M^v 。

基于此, 我们在候选属性集合 C 中, 寻找能够使得划分后基尼指数最小的属性, 将其作为最优化分属性, 即 $c_* = \arg \min(Gini_index(M, c))$, 其中 $c \in C$ 。

此外, 通过更改先验方法, 我们合并了因错误分类造成的损失, 并设置参数以控制分裂适合度。

(3) 结果的过拟合检验——10 倍交叉验证

生成的检测模型可能存在过拟合或独立于数据集的问题, 因此, 我们采用 10 倍交叉

验证法来进行验证，验证方法执行步骤如下：

- ① 将原始的训练数据集划分为 10 个相等子集，将每个子集作为一个折叠，令其为 f_i ， $i=1,2,\dots,10$ 。
- ② 从 $i=1$ 到 $i=10$
 - i. 将 f_i 保留为验证集，并将其余的 9 次折叠保留至交叉验证训练集；
 - ii. 通过交叉验证训练集来训练生成模型，并依据验证集验证预测结果，衡量模型的准确性。
- ③ 计算 10 个验证案例的平均准确性，进一步评估模型准确性。

6.3 任务四模型求解

6.3.1 过拟合检验

根据图 6-2 所示，我们所构建的决策树分类方法能够对测试数据进行较好的检验，模型不存在过拟合问题。

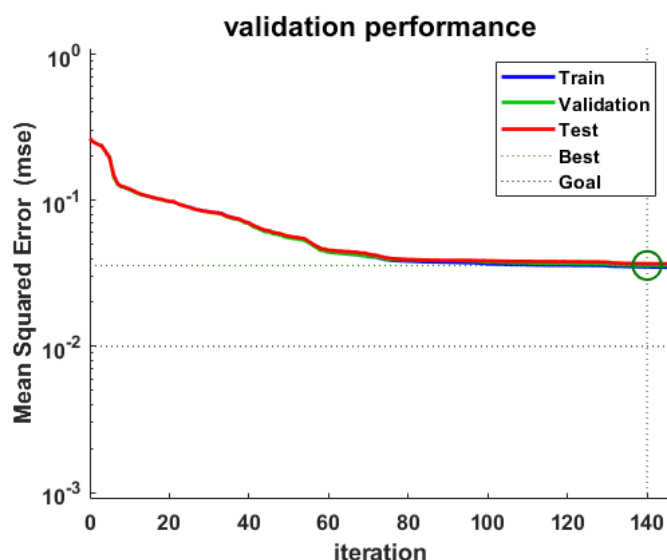


图 6-2 过拟合检验

6.3.2 分类模型的准确性

我们通过决策树学习构建了智能的信号干扰检测方法，将异常数据标记为 0，将正常数据标记为 1。样本数据由附件 1 的 324 个“正常数据”文件和 324 个“异常数据”文件构成，共 138807 个样本用于分类过程，这些样本被随机打乱，以防止模型对特定位置的过拟合。

在附件 1 经预处理的数据中，所构建基于决策树的分类模型对信号存在干扰和不存在干扰的正确检测率分别为 92.1% 和 98.2%，平均准确率达到 95.1%，其混杂矩阵如图 6-3 所示：

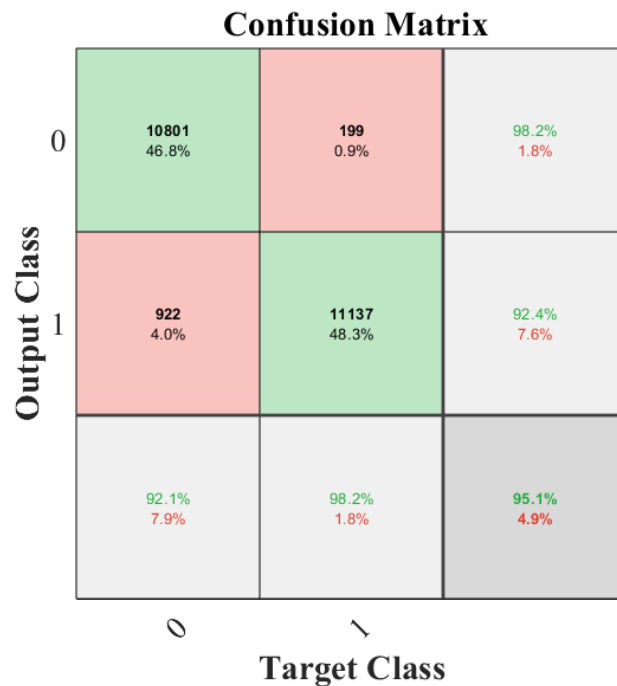


图 6-3 决策树分类结果准确性展示

6.3.3 附件 4 的信号干扰检测

根据所建立的决策树分类模型，对附件 4 中的 10 组数据进行信号干扰判断，结果如表 6-1 所示：

表 6-1 模型对附件四 10 组数据的信号干扰判断

	存在信号干扰	不存在信号干扰
数据集	数据 2、数据 3、数据 4、数据 6、数据 8、数据 9	数据 1、数据 5、数据 7、数据 10

由表 6-1 所示，数据 2、3、4、6、8、9 共 6 个数据存在信号干扰，数据 1、5、7、10 共四个数据不存在信号干扰。

7 问题五建模与求解

7.1 任务五问题分析

在前四个任务中，建立了静态靶点的精确定位模型，当测量距离异常时，能在识别出异常值的基础上，对异常数据进行校正，从而实现整个三维空间上的室内精确定位。对于静态的靶点，测出它与各锚点间的距离并不是一件难事。但在实际生活中，这种应用局限性很小。因此在任务五中，需要构造动态定位模型，精确定位出靶点的运动轨迹。若能成功构建出此动态定位模型，则可以广泛应用到实际生活中，如智慧养老^[7]、光纤传感网络异常节点快速定位。同时，考虑到室内环境中，锚点常常会被墙体、家居等障碍物遮挡，因此锚点与靶点间的信号传播会受到不同程度的影响，在 TOF 方法中具体表现为测量时间变长，导致测得距离值偏大。在靶点的运动中，这种遮挡往往是随机性的、且无法避免的，这对精确定位产生很大的干扰。

本题需在前文静态定位模型的基础上，将其推广至动态定位问题中。动态问题可视为静态问题在每个离散时间节点静态模型的叠加，当时间间隔足够小时，静态模型就转化为了动态模型。因此我们建立 EKF 滤波预测模型来对运动轨迹进行预测，考虑到异常数据的随机性和不可避免性，将此模型进一步优化，构建了自适应 EKF 滤波模型。题目给出，锚点与靶点之间每隔 0.2-0.3 秒发送接收信号一次，而这 0.2-0.3 秒之间的靶点的运动轨迹无法通过模型计算得到，只能通过已知轨迹进行平滑化处理，得到最终完整的运动轨迹图。

本题的具体流程图如图 7-1 所示：

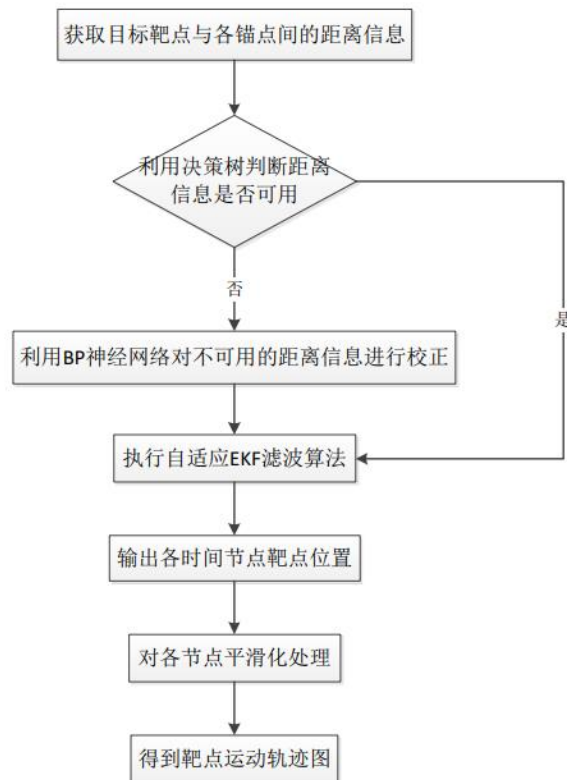


图 7-1 动态定位流程图

主要利用自适应 EKF 滤波算法对位置误差进行预估，预估过程中实时判断 UWB 测量得到的靶点与锚点间的距离信息是否异常，如果出现异常，则对该位置的实际位置进行评估。

7.2 基于自适应 EKF 滤波的定位预测模型

7.2.1 具有数据缺失的动态定位问题

在前文中，我们讨论了 UWB 的静态定位问题，从一定程度上实现了靶点的精确定位。下面，我们进一步考虑 UWB 在一段连续时间上的动态定位问题。首先，在连续的离散时间点上建立了自适应 EKF 滤波预测模型。

EKF 滤波模型

由上文可知，基于 UWB 的精确定位模型是一个非线性系统，每一个时间节点 $t+1$ 的状态受其前一时间节点 t 的影响。因此采用 EKF 滤波算法，将非线性模型在估计值附近泰勒展开，只保留一阶形式，用截断后的一阶近似项作为原方程的近似表达形式，成功地将非线性模型转化成线性模型。考虑测量过程中系统内部存在过程噪声，观察存在观察噪声，可将状态方程表示为：

$$x_{t+1} = f(x_t) + w_t \quad (7.1)$$

测量方程表示为：

$$y_t = h(x_t) + v_t \quad (7.2)$$

其中 t 表示离散时间， x_t 表示系统的状态向量， y_t 表示测量输出。现假设 $f(x_t)$ 和 $h(x_t)$ 对每一时刻的 v_t 都连续可微。 w_t 是系统干扰， v_t 是测量噪声，两者都服从均值为 0 的正态分布，且相互独立，记 w_t, v_t 协方差分别为 Q, R ，由正态分布的性质可知， $Q > 0, R > 0$ 。定义 $f(x_t)$ 和 $h(x_t)$ 的 Jacobi 矩阵分别为 $F_t = \nabla f | \hat{x}_t$ 和 $H_t = \nabla h | \hat{x}_{t|t-1}$ 。

假设下面初始条件成立：

$$\begin{cases} E[x_0] = \hat{x}_0, \\ E[x_t v_m^T] = 0, \\ E[x_t w_m^T] = 0, \\ E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] = P_0. \end{cases} \quad (7.3)$$

其中， $E(x)$ 表示随机变量 x 的数学期望。

EKF 滤波算法主要由时间更新和状态更新两部分构成，状态更新又称为测量更新。每一部分都满足一定的后验关系式：

Step 1 时间更新

$$\begin{cases} \hat{x}_{t+1|t} = f(\hat{x}_t), \\ P_{t+1|t} = F_t P_t A_t^T + Q. \end{cases} \quad (7.4)$$

Step 2 状态更新

$$\begin{cases} K_{t+1} = P_{t+1|t} H_{t+1}^T (H_{t+1} P_{t+1|t} H_{t+1}^T + R)^{-1}, \\ \hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1} (y_{t+1} - h(\hat{x}_{t+1|t})), \\ P_{t+1|t+1} = (I - K_{t+1} H_{t+1}) P_{t+1|t}. \end{cases} \quad (7.5)$$

其中, I 为单位矩阵, F_{t+1}, H_{t+1} 都依赖于前一时间节点的状态估计值, K_{t+1} 表示 EKF 在 $t+1$ 时刻的误差增益矩阵, $\hat{x}_{t+1|t}$ 表示由 t 时刻到 $t+1$ 时刻预测的状态向量, $P_{t+1|t}$ 表示 EKF 由 t 时刻到 $t+1$ 时刻的最小预测误差协方差矩阵, 其也依赖于前一时间节点。

自适应 EKF 滤波预测模型

由于在现实测量中, 测量得到的距离可能存在异常, 无法直接应用, 因此引入了判断变量 α_i^j 判断第 i 个距离信息是否可用。若第 i 个距离信息不可用, 则令 $\alpha_i^j = 0$, 对不可用的距离信息利用 BP 神经网络进行预估, 以保证滤波器的正常运行, 最终实现对靶点位置的实时预估。并利用矩阵 $h(X_{t|t-1})$ 元素 $(i,1)$ 处的值代替不可用的距离信息^[8]。

由此构建出了自适应预测 EKF 滤波模型, 其状态方程为:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ v_{x_t} \\ v_{y_t} \\ v_{z_t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ v_{x_{t-1}} \\ v_{y_{t-1}} \\ v_{z_{t-1}} \end{bmatrix} + w_{t-1} \quad (7.6)$$

其中, $\begin{bmatrix} x_t & y_t & z_t & v_{x_t} & v_{y_t} & v_{z_t} \end{bmatrix}^T$ 表示 t 时刻靶点分别沿 x, y, z 轴正方向的位置和速度, T 为采样周期。

测量方程为:

$$\begin{bmatrix} d_{0,t} \\ d_{1,t} \\ d_{2,t} \\ d_{3,t} \end{bmatrix} = \begin{bmatrix} \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2} \\ \sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2} \\ \sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2} \\ \sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2} \end{bmatrix} + v_t \quad (7.7)$$

其中, $d_{i,t}, t=0,1,2,3$ 为 t 时刻靶点与各个锚点间的测量距离。

对不可用数据进行处理后, 自适应 EKF 滤波预测观测方程变为:

$$\begin{bmatrix} d_{0,t} \\ d_{1,t} \\ d_{2,t} \\ d_{3,t} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (7.8)$$

若 $y_i, i=0,1,2,3$ 数据可用, 则 $y_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$; 若数据不可用, 则 $y_i = h(x_{t|t-1})[i,1]$ 。

利用自适应 EKF 滤波预测，对位置误差进行预估，则式(7.4)(7.5)同步改变，并加入对距离的限制，如下：

$$\begin{cases} K_{t+1} = P_{t+1|t} H_{t+1}^T (H_{t+1} P_{t+1|t} H_{t+1}^T + R_{t+1})^{-1}, \\ \hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1} (y_{t+1} - h(\hat{x}_{t+1|t})), \\ P_{t+1} = (I - K_{t+1} H_{t+1}) P_{t+1|t}, \\ v_{t+1} = y_{t+1} - h(\hat{x}_{t+1|t}) - r_{t+1}, \\ r_{t+1} = (1 - d_{t+1}) r_t + d_t (y_{t+1} - h(\hat{x}_{t+1|t})), \\ R_{t+1} = (1 - d_t) R_t + d_t (v_{t+1} v_{t+1}^T - H_{t+1} P_{t+1|t} H_{t+1}^T), \\ d_t = \frac{1-b}{1-b^{t+1}}, 0 < b < 1. \end{cases} \quad (7.9)$$

其中， r_t, d_t 均为中间误差，Jacobi 矩阵 H_t 为：

$$H_t = \begin{bmatrix} \frac{\partial d_{1,t}}{\partial x} & \frac{\partial d_{1,t}}{\partial y} & \frac{\partial d_{1,t}}{\partial z} & 0 \\ \frac{\partial d_{2,t}}{\partial x} & \frac{\partial d_{2,t}}{\partial y} & \frac{\partial d_{2,t}}{\partial z} & 0 \\ \frac{\partial d_{3,t}}{\partial x} & \frac{\partial d_{3,t}}{\partial y} & \frac{\partial d_{3,t}}{\partial z} & 0 \\ \frac{\partial d_{4,t}}{\partial x} & \frac{\partial d_{3,t}}{\partial y} & \frac{\partial d_{3,t}}{\partial z} & 0 \end{bmatrix}. \quad (7.10)$$

7.2.2 SG 平滑处理

由于锚点与靶点之间每隔 0.2-0.3 秒发送接收信号一次，而这 0.2-0.3 秒之间的靶点的运动轨迹无法通过模型计算得到，只能通过已知轨迹进行平滑化处理，得到最终完整的运动轨迹图。在这里采用了 Savitzky-Golay 滤波（SG 滤波）进行平滑化处理，这是一种基于局部多项式最小二乘法拟合的滤波方法。

将一段曲线的等波长间隔的 5 个点记为集合 X ，SG 平滑算法是利用在各个波长点 $X_{m-2}, X_{m-1}, X_m, X_{m+1}, X_{m+2}$ 的数据的多项式的拟合值替代 X_m ，而后进行移动，直到将曲线遍历。

设滤波窗口宽度 $n = 2m + 1$ ，各个测量点为 $x = (-m, -m+1, 0, \dots, 0, 1, \dots, m-1, m)$ 采用 $k-1$ 次多项式对窗口内的数据进行拟合，表示为：

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_{k-1} x^{k-1} \quad (7.11)$$

n 个这样的方程形成了一个 k 元线性方程组。要使该线性方程组有解，则 $n \geq k$ 。在这里，一般选择 $n > k$ ，通过最小二乘法拟合确定拟合参数 A 。由此得到：

$$\begin{bmatrix} y_{-m} \\ y_{-m-1} \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & -m & \cdots & (-m)^{k-1} \\ 1 & -m+1 & \cdots & (-m+1)^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & m & \cdots & m^{k-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix} + \begin{bmatrix} e_{-m} \\ e_{-m-1} \\ \vdots \\ e_m \end{bmatrix} \quad (7.12)$$

用矩阵表示为:

$$Y_{(2m+1) \times 1} = X_{(2m+1) \times k} A_{k \times 1} + E_{(2m+1) \times 1} \quad (7.13)$$

A 的最小二乘解 \dot{A} 为:

$$\dot{A} = (X^T X)^{-1} X^T Y \quad (7.14)$$

Y 的模型预测值 \dot{Y} 为:

$$\dot{Y} = XA = X(X^T X)^{-1} X^T Y = BY \quad (7.15)$$

其中 $B = X(X^T X)^{-1} X^T$.

7.3 任务五模型求解

首先, 对附件五的数据进行二维表转换, 得到 549 个数据样本, 经过对突变点的识别与剔除, 剩余 500 个样本。对剩余样本进行决策树信号干扰检测, 发现有 219 个“异常数据”, 基于此进行 BP 网络的距离补偿校正, 通过 GN 迭代优化得到异常校正后的靶点坐标。由于本模型定位点的确定仅依赖于此时此刻的静态数据, 在考虑动态轨迹时, 应考虑观测点与相邻时刻数据点的时间关联性。因此, 根据靶点坐标与锚点的距离值计算, 观察四个距离值随时间的变化, 如图 7-2 所示:

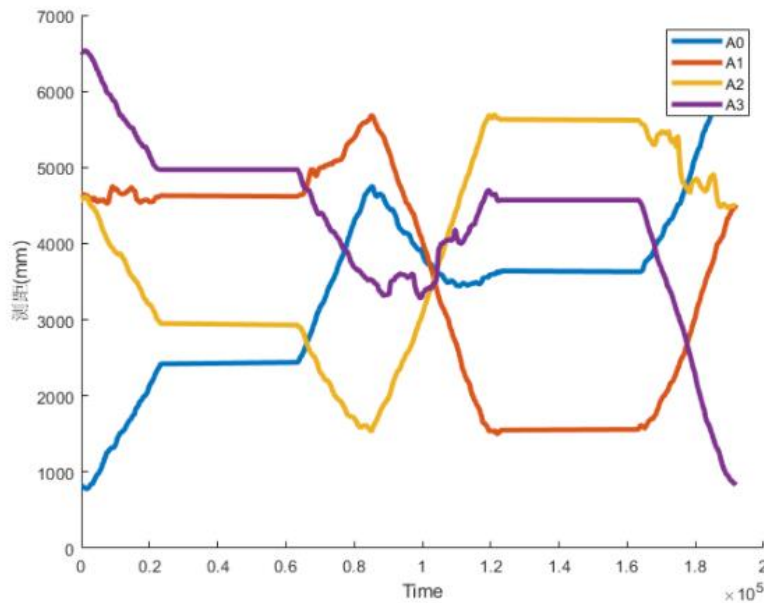


图 7-2 移动靶点与 4 个锚点距离值的时变情况

根据图 7-2 发现，靶点的移动速度并不均匀，与 4 个锚点的距离值在不同的时间呈现不同的急缓变化。此外，靶点的距离值变化从图中明显看出并非平滑，而是存在离散的锯齿段。因此，通过拓展卡尔曼滤波器进行滤波估计，能够得到平滑的运动轨迹，如图 7-3 所示：

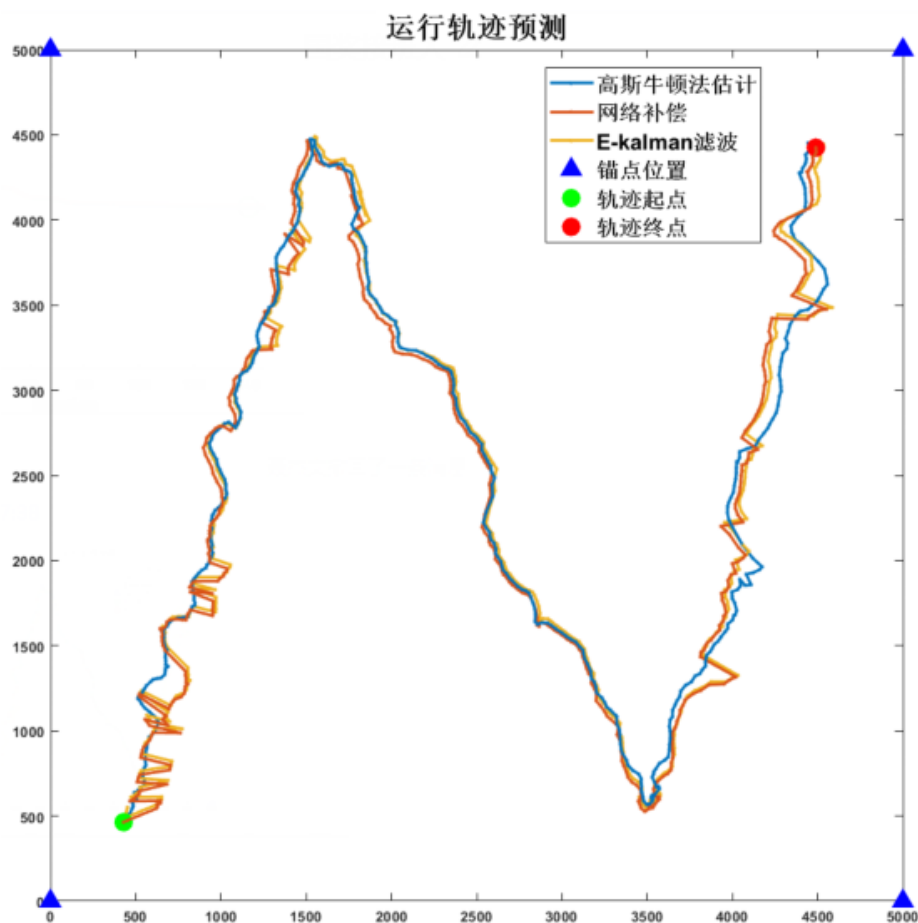


图 7-3 动态靶点的 XY 维度

图 7-3 所示，因为三维坐标在二维平面上展示的局限性，这里画出靶点在 XY 维度上的运动轨迹图，呈现“N”字。可以发现，当 $x \in [1500, 3500]$ 时，三种方法估计的动态轨迹最为接近；而在其他情况时，相较 GN 估计的轨迹，基于 GN-BP 的动态定位轨迹与增加卡尔曼滤波后的轨迹更为贴合。

8 模型评价与推广

8.1 模型评价

8.1.1 模型的优点

(1) 在靶点的精确定位中, 多次利用高斯牛顿迭代法, 减小测量过程中系统误差, 提高定位准确率;

(2) 建立了两个 BP 神经网络模型, 分别对测距正常值和异常值进行处理, 提高了精确定位的准确率:

① 对于正常距离的处理, 利用 BP 神经网络补偿距离的测量误差, 优化了四点定位的精确度;

② 对于异常距离值的处理, 利用 BP 神经网络校正异常数据, 避免了三点定位解的不唯一性, 成功转化为四点定位问题, 定位精度高;

(3) 基于 BP-GN 的精确定位模型能适用于不同规格的室内定位问题, 适应性强;

(4) 利用自适应 EKF 滤波预测模型, 对靶点位置的动态定位进行预测和轨迹追踪, 并对运动轨迹进行平滑化处理, 使得到的运动轨迹图更接近真实运动。

8.1.2 模型的缺点

(1) 在高斯牛顿迭代法中, 由于距离的平方为二阶方程, 本文采用了一阶 Talyor 展开, 此方法主要适用于二阶非线性方程, 所以迭代效果好; 但若处理高阶非线性方程, 则无法产生相同的良好效果, 需要提高 Talyor 展开阶数;

(2) 当靶点的高度超出锚点范围时, 在 z 轴方向上的误差会增大, 但这并不是我们的技术方法造成的。

8.2 模型推广

(1) 若在 UWB 的室内定位中, 增加锚点数量或提高锚点放置高度, 本模型的靶点位置预测准确率将进一步提高;

(2) 基于决策树的信号干扰分类模型可以广泛应用于其他二分类问题中, 结合二叉树的特点, 在处理连续型问题和回归问题方面, 准确率高、计算效率高;

(3) 该定位方法可广泛运用到实际生活中, 如在智慧养老领域, 可对老人的活动进行实时跟踪和监测, 以防出现以外; 在网络故障点定位中, 能在不改变网络内部结构的基础上, 实现故障点的高效精确定位;

(4) 根据异常测距值的分析, 发现其异常的表现形式丰富, 例如, 跳变、渐变。是否可以据此确定干扰情形, 例如移动物体干扰、静态物体干扰等, 将大大提高模型在现实中的应用。

参考文献

- [1] 缪希仁,范建威,江灏,陈静,黄新宇.基站异常情况下基于改进极限学习机的超宽带室内定位方法[J].传感技术学报,2020,33(10):1457-1466.
- [2] 蔡赣飞,徐爱功,洪州,隋心.抗差容积卡尔曼滤波及其在 UWB 室内定位中的应用[J].测绘科学,2018,43(12):123-129.
- [3] Karapistoli, E, Pavlidou, et al. An overview of the IEEE 802.15.4a Standard[J]. IEEE Communications Magazine: Articles, News, and Events of Interest to Communications Engineers, 2010, 48(1):47-53.
- [4] 肖庭杨,章迪.基于 BP 神经网络的厘米级超宽带测距误差改正模型设计与实验[J].全球定位系统,2021,46(04):52-58.
- [5] Musa A , Nugraha G D , H Han, et al. A decision tree - based NLOS detection method for the UWB indoor location tracking accuracy improvement[J]. International Journal of Communication Systems, 2019, 32(13):e3997.1-e3997.13.
- [6] 张少辉,亓玉浩,翟方文,吕洪波,宋亦旭.基于冗余距离筛选的 UWB 定位优化方法[J/OL].清华大学学报(自然科学版):1-9[2021-10-18].<https://doi.org/10.16511/j.cnki.qhdxxb.2021.21.040>.
- [7] Ghorpade Sheetal N.,Zennaro Marco,Chaudhari Bharat S.. IoT-based hybrid optimized fuzzy threshold ELM model for localization of elderly persons[J]. Expert Systems With Applications,2021,184:
- [8] 徐元,赵钦君,程金,张勇,王滨,冯宁,邵丽丽. 面向具有数据缺失 UWB 行人导航的自适应预估 EKF 滤波算法及系统[P]. 山东省: CN109269498B,2020-09-22.

附录

问题一

%数据清洗

```
function [D1,ind_deleted]=Tri_sigma_Filter(D0)
ind_deleted=[];
ind_ori=[1:length(D0)]';
for i=1:10
    Vpp0=max(D0)-min(D0);
    Jump=abs(D0-mean(D0));
    ind_del=find(Jump>3*std(D0));%3 sigma 判据
    D0(ind_del)=[];
    ind_deleted=[ind_deleted;ind_ori(ind_del)];
    ind_ori(ind_del)=[];
    Vpp1=max(D0)-min(D0);
    if Vpp1==Vpp0%无变化时停止
        break
    end
end
D1=D0;
```

问题二

%异常数据补偿 BP 网络

```
%%
clear
load('NET_DATA.mat');
% plot(DATA)
%% 构建，划分输入，输出
P=[DATA_negative.Loc_Est ,DATA_negative.OBSdata]';%估计靶点三坐标，实测四距离，
N*7
T=[DATA_negative.Loc_Real]';%误差估计量
RatioTT=0.85;%训练数据占比
samplenum=size(P,2);%样本数
%% 测试\训练集划分
ind_train=randperm(samplenum,ceil(samplenum*RatioTT));
P_train=P(:,ind_train);%输入，训练
T_train=T(:,ind_train);%输出，训练

ind_test=setdiff([1:samplenum],ind_train);
P_test=P(:,ind_test);%输入，测试
T_test=T(:,ind_test);%输出，测试

%% 构建 BP 网络
%设定参数网络参数
```

```

net = newff(P_train,T_train,20); %隐含层为 20 个神经元
net.trainParam.epochs = 20;
net.trainParam.goal = 1e-7;
net.trainParam.lr = 0.1;

%% 网络训练
net = train(net,P_train,T_train);
negative_net=net;
% save('negative_net.mat','negative_net');
%% 网络测试
T_sim = sim(net,P_test);
% T_sim = sim(net,P_train);

%% 画出误差图
figure
for ii=1:4
    subplot(4,1,ii)
    plot( T_test(ii,:), '-or' )
    hold on
    plot( T_sim(ii,:) , '-*b');
    legend('真实值','预测值')
    R2 = corrcoef(T_sim(ii,:),T_test(ii,:));
    R2 = R2(1,2)^ 2;
    xlabel('预测样本')
    ylabel('strength')
    string = {'BP 网络预测结果对比';[ 'R^2=' num2str(R2)]};
    title(string)
end

%% 计算直接输出靶点坐标的平均误差
LocEst=T_sim'.*para.field;
Loc_real_test=T_test'.*para.field;
load('fitfunz.mat')%拟合函数加载
for ii=1:size(Distance_obs,1)%324
    [err1(ii,:),err2(ii),err3(ii)]=errcal(LocEst(ii,:),Loc_real_test(ii,:));%
    计算一维、二维和三维误差，估计坐标和实际坐标
end
Err1=mean(err1,1)
Err2=mean(err2)
Err3=mean(err3)

%% 计算补偿之后的平均误差
Distance_obs=T_sim'*norm(para.field);
Loc_real_test=T_test'.*para.field;

```

```

load('fitfunz.mat')%拟合函数加载
for ii=1:size(Distance_obs,1)%324
    Dis_obs=fitresult(Distance_obs(ii,:));
    Dis_obs=Distance_obs(ii,:);
    [LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距离，然后牛顿法收敛)
    Err_curve(ii,:)=Err_count(end,:);
    [err1(ii,:),err2(ii),err3(ii)]=errcal(LocEst(ii,:),Loc_real_test(ii,:));%
    计算一维、二维和三维误差，估计坐标和实际坐标
end
Err1=mean(err1,1)
Err2=mean(err2)
Err3=mean(err3)

%% 计算原本的平均误差
Distance_obs=DATA_negative.OBSdata*norm(para.field);
Loc_real_test=DATA_negative.Loc_Real.*para.field;
load('fitfunz.mat')%拟合函数加载
for ii=1:size(Distance_obs,1)%324
    Dis_obs=fitresult(Distance_obs(ii,:));
    %    Dis_obs=Distance_obs(ii,:);
    [LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距离，然后牛顿法收敛)
    Err_curve(ii,:)=Err_count(end,:);
    [err1(ii,:),err2(ii),err3(ii)]=errcal(LocEst(ii,:),Loc_real_test(ii,:));%
    计算一维、二维和三维误差，估计坐标和实际坐标
end
Err1=mean(err1,1)
Err2=mean(err2)
Err3=mean(err3)

% 正常数据补偿网络
clear
load('NET_DATA.mat');
%% 构建，划分输入，输出
P=[DATA_positive.Loc_Est ,DATA_positive.OBSdata]';%估计靶点三坐标，实测四距离，N*7
T=[DATA_positive.Loc_Real]';%误差估计量
RatioTT=0.85;%训练数据占比
samplenum=size(P,2);%样本数
%% 测试\训练集划分
ind_train=randperm(samplenum,ceil(samplenum*RatioTT));

```

```

P_train=P(:,ind_train);%输入，训练
T_train=T(:,ind_train);%输出，训练

ind_test=setdiff([1:samplenum],ind_train);
P_test=P(:,ind_test);%输入，测试
T_test=T(:,ind_test);%输出，测试
%% 构建 BP 网络
%设定参数网络参数
net = newff(P_train,T_train,20); %隐含层为 20 个神经元
net.trainParam.epochs = 20;
net.trainParam.goal = 1e-7;
net.trainParam.lr = 0.1;
%% 网络训练
net = train(net,P_train,T_train);
positive_net=net;
save('positive_net.mat','positive_net');
%% 网络测试
T_sim = sim(net,P_test);
%% 画出误差图
figure
for ii=1:3
    subplot(3,1,ii)
    plot( T_test(ii,:), '-or' )
    hold on
    plot( T_sim(ii,:) , '-*b');
    legend('真实值','预测值')
    R2 = corrcoef(T_sim(ii,:),T_test(ii,:));
    R2 = R2(1,2)^ 2;
    xlabel('预测样本')
    ylabel('strength')
    string = {'BP 网络预测结果对比';['R^2=' num2str(R2)]};
    title(string)
end
%% 计算直接输出靶点坐标的平均误差
LocEst=T_sim'.*para.field;
Loc_real_test=T_test'.*para.field;

Distance_obs=LocEst;
load('fitfunz.mat')%拟合函数加载
for ii=1:size(Distance_obs,1)%324
%     Dis_obs=fitresult(Distance_obs(ii,:));
%     Dis_obs=Distance_obs(ii,:);
%     [LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和
观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个

```

距离，然后牛顿法收敛)

```
% Err_curve(ii,:)=Err_count(end,:);  
[err1(ii,:),err2(ii),err3(ii)]=errcal(LocEst(ii,:),Loc_real_test(ii,:));%  
计算一维、二维和三维误差，估计坐标和实际坐标  
end  
Err1=mean(err1,1)  
Err2=mean(err2)  
Err3=mean(err3)
```

%附件二 10 个坐标值的定位求解

%% 第二问解答，计算十个坐标

clear

%% 数据输入，标准格式，归一化，带着场景信息

```
Bsnum=4;  
Bsloc=[0 0 1300;5000 0 1700;0 5000 1700;5000 5000 1300];  
para.Thr = 100;  
para.DataLen=1;  
para.BsNum=Bsnum;  
para.BsLoc=Bsloc;  
para.MaxIters=100;  
para.xinit = mean(Bsloc,1);  
para.dRef=2500;  
para.field=[5000,5000,3000];
```

```
DATA.OBSdata=[1320 3950 4540 5760;  
3580 2580 4610 3730;  
2930 2600 4740 4420;  
2740 2720 4670 4790;  
2980 4310 2820 4320;  
2230 3230 4910 5180;  
4520 1990 5600 3360;  
2480 3530 4180 5070;  
4220 2510 4670 3490;  
5150 2120 5800 2770;];%前五个无干扰，后五个有干扰
```

```
DATA.PNtag=[ones(1,5),zeros(1,5)];
```

```
load fitfunz
```

%% 靶心直接估计

```
Distance_obs=DATA.OBSdata;
```

```
for ii=1:size(Distance_obs,1)
```

```
Dis_obs=fitresult(Distance_obs(ii,:));
```

```
% Dis_obs=Distance_obs(ii,:);
```

```
[LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和观  
测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距
```

```

离，然后牛顿法收敛)
end
DATA.Loc_Est=LocEst;
%% 归一化数据
DATA.Loc_Est=DATA.Loc_Est./para.field;
DATA.OBSdata=DATA.OBSdata/norm(para.field);
%% 异常判断，由于已知异常正常，这里可以验证异常判断模型准确率，准确率 100%
load('Judge_net.mat')
P_judge=DATA.OBSdata';

T_sim=sim(judge_net,P_judge);
tag=zeros(size(T_sim));

indP=find(T_sim>0.5);%索引有用
indN=find(T_sim<=0.5);
tag(indP)=1;
tag(indN)=0;

DATA.PNtag=tag;%更新 tag，p 为 1，n 为 0
%% 正常数据网络补偿，输出靶心
load('positive_net.mat')
P_positive=[DATA.Loc_Est(indP,:), DATA.OBSdata(indP,:)]';
T_sim=sim(positive_net,P_positive);
Loc_Est1=T_sim'.*para.field;
%% 异常数据网络补偿，输出靶心
load('negative_net.mat')
P_negative=[DATA.Loc_Est(indN,:), DATA.OBSdata(indN,:)]';
T_sim=sim(negative_net,P_negative);
Loc_Est0=T_sim'.*para.field;
%% 组合正负样本数据，恢复顺序
Loc_Est_final=zeros(size(DATA.Loc_Est));
Loc_Est_final(indP,:)=Loc_Est1;
Loc_Est_final(indN,:)=Loc_Est0;
%% 回代验证，计算正常部分的距离误差
Distance_obs=DATA.OBSdata*norm(para.field);
for ii=1:5%324
    Distance_Obs=Distance_obs(ii,:);%观测四距离，N*4
    Distance_Est= UWBTtoObs(para.BsLoc,para.BsNum,Loc_Est_final(ii,:));%估计四
距离
    Err_dis(ii,:)=abs(Distance_Obs-Distance_Est);
end
mean(Err_dis(:))
max(Err_dis(:))
min(Err_dis(:))

```


问题三

% 附件三 10 组数据的精确定位

```
clear
%% 数据输入，标准格式，归一化，带着场景信息
Bsnum=4;
Bsloc=[0 0 1200;
        5000 0 1600;
        0 3000 1600;
        5000 3000 1200];
para.Thr = 100;
para.DataLen=1;
para.BsNum=Bsnum;
para.BsLoc=Bsloc;
para.MaxIters=100;
para.xinit = mean(Bsloc,1);
para.dRef=2500;

DATA.OBSdata=[4220 2580 3730 1450;
              4500 1940 4420 1460;
              3550 2510 3410 2140;
              3300 3130 2900 2790;
              720 4520 3050 5380;
              5100 2220 4970 800;
              2900 3210 3140 2890;
              2380 3530 2320 3760;
              2150 3220 3140 3640;
              1620 3950 2580 4440];%前五个无干扰，后五个有干扰
para.field=[5000,3000,3000];%%换场景范围，归一化依据
DATA.PNtag=[ones(1,5),zeros(1,5)];
load fitfunz
%% 靶心直接估计
Distance_obs=DATA.OBSdata;
for ii=1:size(Distance_obs,1)
    Dis_obs=fitresult(Distance_obs(ii,:));
    [LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距离，然后牛顿法收敛)
end
DATA.Loc_Est=LocEst;
%% 归一化数据
DATA.Loc_Est=DATA.Loc_Est./para.field;
DATA.OBSdata=DATA.OBSdata/norm(para.field);
```

```

%% 正常异常数据划分
indN=[1:10];
%% 正常数据网络补偿, 输出靶心
load('positive_net.mat')
P_positive=[DATA.Loc_Est(indP,:), DATA.OBSdata(indP,:)]';
T_sim=sim(positive_net,P_positive);
Loc_Est1=T_sim'.*para.field;
%% 异常数据网络补偿, 输出靶心
load('negative_net.mat')
P_negative=[DATA.Loc_Est(indN,:), DATA.OBSdata(indN,:)]';
T_sim=sim(negative_net,P_negative);
Loc_Est0=T_sim'.*para.field;
%% 组合正负样本数据, 恢复顺序
Loc_Est_final=zeros(size(DATA.Loc_Est));
Loc_Est_final(indN,:)=Loc_Est0;
%% 回代正常的计算误差
Distance_obs=DATA.OBSdata*norm(para.field);
for ii=1:5%324
    Distance_Obs=Distance_obs(ii,:);%%观测四距离, N*4
    Distance_Est= UWBTtoObs(para.BsLoc, para.BsNum, Loc_Est_final(ii,:));%估计四
    距离
    Err_dis(ii,:)=abs(Distance_Obs-Distance_Est);
end
mean(Err_dis(:))
max(Err_dis(:))
min(Err_dis(:))

```

问题四

%% 信号干扰判断模型

%% 自行判断有无干扰, 调用训练好的判断模型

```

clear
%% 数据输入, 标准格式, 归一化, 带着场景信息
Bsnum=4;
Bsloc=[0 0 1300;5000 0 1700;0 5000 1700;5000 5000 1300];

para.Thr = 100;
para.DataLen=1;

para.BsNum=Bsnum;
para.BsLoc=Bsloc;
para.MaxIters=100;
para.xinit = mean(Bsloc,1);
para.dRef=2500;

```

```

para.field=[5000, 5000, 3000];

DATA.OBSdata=[2940 4290 2840 4190;
5240 5360 2040 2940;
4800 2610 4750 2550;
5010 4120 3810 2020;
2840 4490 2860 4190;
5010 5320 1990 2930;
5050 3740 3710 2070;
5050 4110 3710 2110;
4840 2600 4960 2700;
2740 2720 4670 4790];
DATA.PNtag=[ones(1,5), zeros(1,5)];
load fitfunz
%% 靶心直接估计
Distance_obs=DATA.OBSdata;
for ii=1:size(Distance_obs,1)
    Dis_obs=fitresult(Distance_obs(ii,:));
%    Dis_obs=Distance_obs(ii,:);
    [LocEst(ii,:), Err_count]= UWBLocIters(Dis_obs',para);%输入靶点的坐标和观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距离，然后牛顿法收敛)
end
DATA.Loc_Est=LocEst;
%% 归一化数据
DATA.Loc_Est=DATA.Loc_Est./para.field;
DATA.OBSdata=DATA.OBSdata/norm(para.field);
%% 二三问网络判断，已知异常正常，可以用于验证网络
load('Judge_net.mat')
P_judge=DATA.OBSdata';
T_sim=sim(judge_net,P_judge);
tag=zeros(size(T_sim));
%阈值二分标签
indP=find(T_sim>0.5);%索引
indN=find(T_sim<=0.5);
tag(indP)=1;
tag(indN)=0;
DATA.PNtag=tag%更新 tag, p 为 1, n 为 0

```

问题五

%卡尔曼滤波

% GPS 精度为 1m、气压计精度为 0.5m，加速度计的精度为 1cm/s^2

```

% 无人机按照螺旋线飞行，半径为 20m，螺距为 40m，100s 完成一圈飞行
% 数据采集频率为 10Hz
%
% clear all
function Loc_Est_KF =KF(Loc_Est_inter,para)

D = 3; % 维度，可取 1,2,3

N =size(Loc_Est_inter,1); % 采样点数

A = eye(D); % 状态转移矩阵，和上一时刻状态没有换算，故取 D
阶单位矩阵
A(1,2)=100;
A(2,3)=100;
x = zeros(D, N); % 存储滤波后的数据
% z = ones(D, N); % 存储滤波前的数据
x(:, 1) = Loc_Est_inter(1,:)' ; % 初始值设为 1（可为任意数）
P = eye(D); % 初始值为 1（可为非零任意数），取 D 阶单位矩阵

Q = 5*eye(D); % 过程噪声协方差，估计一个
R = 5*eye(3); % 测量噪声协方差，精度为多少取多少

k = 1; % 采样点计数

% 三维 x,y,z 方向，GPS 和气压计
% true3D = [r * cos(w*t0); r * sin(w*t0); t0 * 0.4];

for t = 1:size(Loc_Est_inter,1)
    k = k + 1;
    x(:,k) = A * x(:,k-1); % 卡尔曼公式 1
    P = A * P * A' + Q; % 卡尔曼公式 2
    H = eye(D);
    K = P*H' * inv(H*P*H' + R); % 卡尔曼公式 3
    % 三维 x,y,z 方向
    z(:,k) =Loc_Est_inter(t,:)' ;%滤波器前数据

    x(:,k) = x(:,k) + K * (z(:,k)-H*x(:,k)); % 卡尔曼公式 4
    P = (eye(D)-K*H) * P; % 卡尔曼公式 5
end
Loc_Est_KF=x;
%% 三维情况
% figure

```

```

% plot3(z(1,:),z(2,:),z(3,:), 'b. '); % 绘制滤波前数据
% axis('equal');grid on;hold on % 坐标等距、继续绘图、添加网格
% plot3(x(1,:),x(2,:),x(3,:), 'r. '); % 绘制滤波后数据
% plot3(true3D(1,:), true3D(2,:), true3D(3,:)); % 绘制滤波后数据
% legend('滤波前','滤波后','理想值'); % 绘制真实值
% xlabel('x 方向: m');
% ylabel('y 方向: m');
% zlabel('高度: m');hold off;

```

%% 靶点动态轨迹定位

```

clear
%% 数据输入，标准格式，归一化，带着场景信息，
Bnum=4;
Bsloc=[0 0 1300;5000 0 1700;0 5000 1700;5000 5000 1300];
para.Thr = 100;
para.DataLen=1;
para.BsNum=Bnum;
para.BsLoc=Bsloc;
para.MaxIters=100;
para.xinit = mean(Bsloc,1);
para.dRef=2500;
para.field=[5000, 5000, 3000];

[OBSdata,t]=filereadl_modi('附件 5：动态轨迹数据.txt',Bnum,1,[]);
DATA.OBSdata=OBSdata;
load fitfunz
%% 靶心直接估计
Distance_obs=DATA.OBSdata;
for ii=1:size(Distance_obs,1)
    Dis_obs=fitresult(Distance_obs(ii,:));
    [LocEst(ii,:), Err_count]= UWBLocIters1(Dis_obs',para);%输入靶点的坐标和观测到的四个距离，迭代出误差最小的估计坐标(不断更新估计坐标，计算和锚点的四个距离，然后牛顿法收敛)
end
DATA.Loc_Est=LocEst;
%% 归一化数据
DATA.Loc_Est=DATA.Loc_Est./para.field;
DATA.OBSdata=DATA.OBSdata/norm(para.field);

%% 异常点判断
load('Judge_net.mat')
P_judge=DATA.OBSdata';

```

```

T_sim=sim(judge_net,P_judge);
tag=zeros(size(T_sim));
%阈值分隔标签
indP=find(T_sim>0.5);%
indN=find(T_sim<=0.5);
tag(indP)=1;
tag(indN)=0;
DATA.PNtag=tag;%更新 tag, p 为 1, n 为 0
%% 正常数据网络补偿, 输出靶心
load('positive_net.mat')
P_positive=[DATA.Loc_Est(indP,:), DATA.OBSdata(indP,:)]';
T_sim=sim(positive_net,P_positive);
Loc_Est1=T_sim'.*para.field;
%% 异常数据网络补偿, 输出靶心
load('negative_net.mat')
P_negative=[DATA.Loc_Est(indN,:), DATA.OBSdata(indN,:)]';
T_sim=sim(negative_net,P_negative);
Loc_Est0=T_sim'.*para.field;
%% 组合正负样本数据, 恢复顺序
Loc_Est_final=zeros(size(DATA.Loc_Est));
Loc_Est_final(indP,:)=Loc_Est1;
Loc_Est_final(indN,:)=Loc_Est0;% 神经网络补偿之后的估计值
%% kalman 滤波
%% 计算四个值
Distance=[];
for ii=1:size(Loc_Est_final,1)
    Distance_temp= UWBToaObs(Bsloc,Bsnum,Loc_Est_final(ii,:));%实际四距离
    Distance=[Distance; Distance_temp];%和观测距离数目相等, 每个文件相同
end
%% 时间连续, 等时间间隔插值
T2=t-t(1);
dt=T2(end)/500;
tseq=0:dt:T2(end);
for ii=1:Bnum
    OBS_inter(:,ii)=interp1(T2, Distance(:,ii),tseq,'linear');
end
%% 卡尔曼滤波器
Loc_Est_KF =KF(Loc_Est_final,para)';
%% 绘制轨迹图
figure;
hold on;
plot3(LocEst(:,1),LocEst(:,2),LocEst(:,3),'.-','LineWidth',2);
plot3(Loc_Est_final(:,1),Loc_Est_final(:,2),Loc_Est_final(:,3),'.-','LineWidt
h',2);

```

```

plot3(Loc_Est_KF(:,1),Loc_Est_KF(:,2),Loc_Est_KF(:,3),'.-','LineWidth',2);
plot3(Bsloc(:,1),Bsloc(:,2),Bsloc(:,3),'b^','LineWidth',5);
legend('直接估计','直接估计+网络补偿','直接估计+网络补偿+E-kalman 滤波','锚点
位置');
title('运行轨迹预测')
axis([0 5e3 0 5e3 0 3e3]);

```